

AD-A218 179

Ada Compiler Performance Test Suite and Test
Evaluation Capability (ACPS) User's Guide

Prepared by

R. E. KAYFES
Information Processing Division
Engineering Group
The Aerospace Corporation
El Segundo, CA 90245-4691

DTIC
ELECTE
FEB 21 1990
S D

September 1988

Prepared for

SPACE SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
Los Angeles Air Force Base
P.O. Box 92960
Los Angeles, CA 90009-2960

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

93 02

7

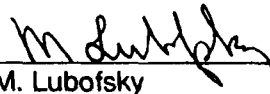
This report was submitted by The Aerospace Corporation, El Segundo, CA 90245, under Contract No. F04701-88-C-0089 with Space Systems Division, P.O. Box 92960, Los Angeles, CA 90009-2960. It was reviewed and approved for The Aerospace Corporation by R. D. Hefner, Director, Software Development Department, Engineering Group. Capt. John Brill, SSD/ALR, was the project officer for the program.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.



A. E. Stevens, Lt. Col., USAF
SSD/ALR



M. Lubofsky
Senior Engineering Specialist
Computer Resources Management
and Standards Office

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; Distribution unlimited		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) TR-0089(4902-03)-2			5. MONITORING ORGANIZATION REPORT NUMBER(S) SSD-TR-89-83		
6a. NAME OF PERFORMING ORGANIZATION Engineering Group The Aerospace Corp.		6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Air Force Systems Command Space Systems Division		
6c. ADDRESS (City, State, and ZIP Code) 2350 E. El Segundo Blvd El Segundo, CA 90245			7b. ADDRESS (City, State, and ZIP Code) Los Angeles Air Force Base Los Angeles, CA 90009-2960		
8a. NAME OF FUNDING / SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F04701-88-C-0089		
8c. ADDRESS (City, State, and ZIP Code)			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) Ada Compiler Performance Test Suite and Test Evaluation Capability (ACPS) User's Guide					
12. PERSONAL AUTHOR(S) R.E. Kayfes					
13a. TYPE OF REPORT		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Year, Month, Day) September 1988	
				15. PAGE COUNT 313	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Ada compiler performance, Ada run-time environments		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
<p>The purpose of this document is to describe the set of software tests which is designed to assist users in evaluating the performance of run-time environments provided by Ada compilation systems. The Ada computer programs, primarily designed and written at The Aerospace Corporation, and augmented by JOVIAL and FORTRAN benchmarks, provide tools to display the results of test executions and comparisons among the three languages. Output from the tests includes compile-time and run-time statistics, such as elapsed time, CPU time, code and data sizes, and virtual and physical memory usage. This document describes how to (a) install the test suite, (b) set up the test environment, (c) execute the tests, and (d) compare the test results for executions on different compilers or for those on the same compiler using different compilation options.</p>					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Lt. Col. A.E. Stevens			22b. TELEPHONE (Include Area Code) (213) 643-2532		22c. OFFICE SYMBOL SSD/ALR

EXECUTIVE SUMMARY

The Ada Compiler Performance Test Suite and Test Evaluation Capability (ACPS) is a collection of programs designed to assist users in evaluating the performance of run-time environments provided by Ada compilation systems. A unique feature of the ACPS is that it provides an extensive set of JOVIAL and FORTRAN benchmarks to aid assessment of the likely performance impact of transitioning programs from JOVIAL or FORTRAN to Ada. The ACPS test suite includes programs to test both single language features (e.g., type conversion) and multiple language features (e.g., the Whetstone benchmark) written in Ada, JOVIAL, and FORTRAN. The suite also provides software to gather and report performance statistics in a format common to all three test languages.

The Ada programs of the ACPS are written in standard Ada (ANSI/MIL-STD-1815A), were designed to be easily rehostable, and are not restricted to a particular machine environment. The user's guide supplied with the ACPS describes the steps necessary to rehost the ACPS and identifies the types of source code modifications that may be necessary when applying the ACPS to a different compiler.

The ACPS also provides tools to display the results of test executions, and to automatically compare corresponding results from Ada, JOVIAL, and FORTRAN test executions. Tool output includes compile-time and run time statistics such as elapsed time, CPU time, code and data sizes, virtual and physical memory usage, etc., in order to facilitate complete and objective comparisons.

The ACPS is available on tape in either Digital Equipment Corporation (DEC) VAX/VMS BACKUP format or in ANSI tape format for users wishing to rehost the ACPS. The user should have a working understanding of the host operating system and command language if the ACPS is to be installed on a system other than DEC VAX/VMS.

The ACPS is primarily intended for users who are familiar with Ada and who wish to get detailed information on the run-time efficiency of an Ada implementation for a particular application. The ACPS is not intended to be the only instrument for measuring the efficiency of Ada run-time environments. Performance issues and selection criteria for Ada run-time environments must be developed in accordance with the requirements of specific applications. The primary benefit of using the ACPS will be in its support of applications planning. It can be used to demonstrate which features of Ada are inefficiently implemented through comparison of equivalent Ada, JOVIAL, and FORTRAN test program executions. This information can be used in several ways. It can be used as justification for avoiding certain language features or processing options. It can focus attention to those areas of the compiler that must be improved and provide a mechanism for measuring improvements or degradations in run-time performance as a compiler implementation evolves. The ACPS can also be used to assist determination as to whether existing JOVIAL or FORTRAN modules should be converted to Ada by showing the probable performance impact of straightforward translation to Ada as would likely be obtained through use of a language translation tool.

ACKNOWLEDGMENTS

This report is sponsored by Space Systems Division's Directorate for Computer Resources (SSD/ALR). Funding for the effort was provided by the Air Force Computer Resource Management Technology Program, Program Element (PE) 64740F, Project 2526, Software Engineering Tools and Methods.

Program Element 64740F is the Air Force engineering development program to develop and transfer into active use the technology, tools, and techniques needed to cope with the explosive growth in Air Force systems that use computer resources. The goals of the program are to: (a) provide for the transition of computer system developments from laboratories, industry, and academia to Air Force systems; (b) develop and apply software acquisition management techniques to reduce life-cycle costs; (c) provide improved software design tools; (d) address the various problems associated with computer security; (e) develop advanced software engineering tools, techniques, and systems; (f) support the implementation of high-order languages (e.g., Ada); (g) address human engineering for computer systems; and (h) develop and apply computer simulation techniques for the acquisition process.

The Ada Compiler Performance Test Suite and Test Evaluation Capability was developed at The Aerospace Corporation by the following team of software engineers: Allen R. Adams, Kathleen M. Bruckert, Richard E. Kayfes, Michael J. McLaughlin, and Beverly L. Parmelee. Team members involved in development of each software module are noted by initials in commentary prologue in each module.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Availability Codes
A-1	

DTIC
COPY
INSPECTED

CONTENTS

	EXECUTIVE SUMMARY.....	1
	ACKNOWLEDGMENTS.....	3
1.	INTRODUCTION.....	1-1
1.1	Test Suite.....	1-3
1.2	Test Evaluation Capability.....	1-7
2.	TEST CATEGORIES.....	2-1
3.	TEST MEASUREMENT CONCERNS.....	3-1
3.1	Operating System Effect.....	3-3
3.2	Hardware Effect.....	3-4
4.	TEST MEASUREMENT AND CODING TECHNIQUES.....	4-1
4.1	Compile-Time Measurement Techniques.....	4-1
4.2	Run-Time Measurement Techniques.....	4-1
4.3	ACPS Coding Technique Examples.....	4-3
5.	ACPS EVALUATION TESTING SOFTWARE ARCHITECTURE.....	5-1
6.	NAMING CONVENTIONS.....	6-1
6.1	File Name Format.....	6-2
6.2	Test Name Format.....	6-3
7.	ACPS INSTALLATION AND TEST EXECUTION.....	7-1
7.1	Installation of ACPS Source Files.....	7-1
7.2	Creation of Command Procedures.....	7-2
7.3	Machine-Dependent Software Modifications.....	7-3
7.3.1	Machine-Specific Test Support Software.....	7-3
7.3.2	Language-Specific Test Support Software.....	7-3
7.3.3	ACPS Tests.....	7-5
7.3.4	ACPS Test Comparison Tool.....	7-7
7.4	Determination of Test Duration.....	7-7
7.5	Determination of Test Measurement Accuracy.....	7-7
7.6	Performance of Test Comparisons.....	7-8
8.	Ada RUN-TIME ENVIRONMENT PERFORMANCE EVALUATION.....	8-1
APPENDICES:		
A.	ANSI TAPE FORMAT.....	A-1
B.	VAX/VMS BACKUP TAPE FORMAT.....	B-1

CONTENTS (Continued)

APPENDICES (Continued):

C.	VAX/VMS SAMPLE COMMAND PROCEDURES.....	C-1
C.1	Installation of ACPS Source and Command Files.....	C-1
C.2	Installation of Common Language Support Software.....	C-2
C.3	Installation of ACPS Comparison Tools.....	C-2
C.4	Determination of VAX/VMS Software Environment.....	C-2
C.5	Determination of Ada Compiler-Dependent Parameters.....	C-3
C.6	ACPS Command Procedure Techniques.....	C-4
C.7	Determination of Test Loop Overhead Parameters.....	C-6
C.8	Determination of Test Repeatability.....	C-6
C.9	Compilation and Execution of ACPS Tests.....	C-7
C.10	Execution of ACPS Test Result Comparison Tool.....	C-8
D.	Ada TEST INTERFACE PACKAGE SPECIFICATIONS.....	D-1
E.	ACPS TEST COMPARISON TOOL.....	E-1
E.1	ACPS Compile-Time Test Result Comparator.....	E-1
E.1.1	Compile-Time Test Result Input File Formats.....	E-1
E.1.2	User Input.....	E-4
E.1.3	CCOMP Output File Formats.....	E-4
E.2	ACPS Run-Time Test Result Comparator.....	E-6
E.2.1	Run-Time Test Result Input File Formats.....	E-6
E.2.2	User Input.....	E-10
E.2.3	CEXEC Output File Formats.....	E-11
E.2.4	Measurement of Test Repeatability.....	E-14
E.3	Test Name and File Name Order.....	E-15
F.	ACPS TEST DESCRIPTIONS.....	F-1
G.	TEST PROGRAMS AND SOURCE CODE FILES.....	G-1

FIGURES

1-1	Ada Run-Time Environment Components.....	1-2
1-2	Ada Test Suite Generation.....	1-4
1-3	JOVIAL Test Suite Generation.....	1-5
1-4	FORTRAN Test Suite Generation.....	1-6
5-1	ACPS Test Evaluation Architecture.....	5-3
5-2	Compile-Time Evaluation Architecture.....	5-4
5-3	Run-Time Evaluation Architecture.....	5-5
5-4	Test Compilation Comparator Interfaces.....	5-6
5-5	Test Execution Comparator Interfaces.....	5-7

TABLES

4-1	Ada Test Loop Overhead Program.....	4-8
4-2	Sample Test Iteration Count Package.....	4-9
4-3	JOVIAL Test Loop Overhead Program.....	4-9
4-4	FORTRAN Test Loop Overhead Program.....	4-10
4-5	Example IF Statement Test Overhead Package.....	4-11
4-6	Example IF Statement Test.....	4-12
4-7	Example IF Statement Test Driver.....	4-13
6-1	ACPS File Types.....	6-5
7-1	Ada Type A Test Input/Output Files.....	7-10
7-2	FORTRAN Test Program Input/Output Files.....	7-11
7-3	Test Support Software File Types.....	7-12
7-4	Language-Specific Test Support Software Files.....	7-13
A-1	ACPS ANSI Tape Contents Volume ACPS.....	A-3
A-2	ACPS ANSI Tape Contents Volume ACPS02.....	A-20
A-3	ACPS ANSI Tape Contents Volume ACPS03.....	A-25
A-4	ACPS ANSI Tape Contents Volume ACPS04.....	A-29
A-5	ACPS ANSI Tape Contents Volume ACPD05.....	A-33
C-1	ACPS VAX/VMS Command Procedure.....	C-10
C-2	Command Procedure DACPS:LOGICALS.COM.....	C-11
C-3	Command Procedure DSUP:SUPPORT.COM.....	C-12
C-4	Command Procedure DTOOLS:TOOLS.COM.....	C-13
C-5	Command Procedure DACPS:VAXVMS.COM.....	C-14
C-6	Command Procedure DDEC:ADAPARM.COM.....	C-15
C-7	ADAPARM Output for DEC VAX Ada.....	C-17
C-8	Command Procedure DDEC:SETUP.COM.....	C-18
C-9	Command Procedure DSUP:SYMBOLS.COM.....	C-19
C-10	Command Procedure DSUP:RUN.COM.....	C-20
C-11	Command Procedure DSUP:EXEC.COM.....	C-21
C-12	Command Procedure DSUP:SHOW.COM.....	C-22
C-13	Command Procedure DSUP:SHOW_.COM.....	C-23
C-14	Command Procedure DDEC:AFIRST.COM.....	C-24
C-15	AFIRST Output for DEC VAX Ada.....	C-26
C-16	Command Procedure DDEC:REPEAT.COM.....	C-28
C-17	Command Procedure DDEC:ABATCH.COM.....	C-29
C-18	Command Procedure DDEC:ACOMP.COM.....	C-31
C-19	Command Procedure DDEC:ACOM.COM.....	C-34
C-20	Command Procedure DDEC:ALNK.COM.....	C-35
C-21	Command Procedure DDEC:ASRC.COM.....	C-36
C-22	Command Procedure DDEC:AOBJ.COM.....	C-37
C-23	Command Procedure DDEC:AEEXE.COM.....	C-38
C-24	Command Procedure DDEC:AEEXEC.COM.....	C-39
C-25	Command Procedure DTOOLS:CCOMP.COM.....	C-40
C-26	Command Procedure DTOOLS:CEXEC.COM.....	C-41

TABLES (Continued)

D-1	OURSYS Package Specification.....	D-3
D-2	OURSYS Package Body Declarations.....	D-9
D-3	OURSPC Package Specification.....	D-14
D-4	OURTYP Package Specification.....	D-19
D-5	OURDMP Package Specification.....	D-21
D-6	MATHFUN Package Specification.....	D-21
E-1	CCOMP Interface File Specifications.....	E-16
E-2	CEXEC Interface File Specifications.....	E-17
E-3	DEC VAX Ada Compile-Time Test Statistic File.....	E-18
E-4	ECSP0 JOVIAL Compile-Time Test Statistic File.....	E-21
E-5	ACPS Compilation Comparator Formatted Ada File.....	E-24
E-6	ACPS Compilation Comparator Formatted JOVIAL File.....	E-27
E-7	ACPS Compilation Comparator Comparison Output File.....	E-30
E-8	DEC VAX Ada Run-Time Test Statistic File.....	E-36
E-9	ECSP0 JOVIAL Run-Time Test Statistic File.....	E-37
E-10	CEXEC Formatted VAX Ada Input File without VMS Statistics.....	E-38
E-11	CEXEC Formatted ECSP0 JOVIAL Input File without VMS Statistics.....	E-39
E-12	CEXEC Comparison Output File without VMS Statistics.....	E-40
E-13	CEXEC Formatted VAX Ada Input File with VMS Statistics.....	E-42
E-14	CEXEC Formatted ECSP0 JOVIAL Input File with VMS Statistics...	E-43
E-15	CEXEC Comparison Output File with VMS Statistics.....	E-44
F-1	ACPS Test Descriptions.....	F-2
G-1	Ada Type A Test Programs and Source Code Files.....	G-2
G-2	Ada Type C Test Programs and Source Code Files.....	G-11
G-3	Ada Type E Test Programs and Source Code Files.....	G-19
G-4	Ada Type S Test Programs and Source Code Files.....	G-27
G-5	Ada Type T Test Programs and Source Code Files.....	G-36
G-6	JOVIAL Test Programs and Source Code Files.....	G-45
G-7	FORTTRAN Test Programs and Source Code Files.....	G-53

1. INTRODUCTION

The Ada Compiler Performance Test Suite and Test Evaluation Capability (ACPS) is a set of programs designed to assist users in evaluating the performance of run-time environments provided by Ada compilation systems. The ACPS is primarily composed of test cases designed and coded at The Aerospace Corporation. The ACPS has also been augmented by public domain benchmarks where appropriate. These tests have been modified to run in the ACPS test environment and annotated to identify their origin. The ACPS tests were designed to execute within the framework of a test evaluation system consisting of test coding techniques, test result reporting conventions, and tools for automated test result comparisons. The VAX/VMS version of the ACPS test evaluation system is delivered on tape in VAX/VMS BACKUP format. A rehostable version of the ACPS is delivered in ANSI tape format.

The purpose of this document is to describe how to (a) install the ACPS test suite, (b) set up the ACPS test environment, (c) execute the tests, and (d) automatically compare the test results for executions on different compilers or for executions on the same compiler using different compilation options. In addition, this document provides guidance on how to make use of the ACPS test results in evaluation of Ada run-time environments.

The primary objective of the ACPS test suite is to provide information useful to the evaluation of Ada run-time environments for real-time space applications. In our view, an Ada run-time environment is composed of the entire hardware/software configuration that permits execution of Ada-compiled programs. Figure 1-1 depicts the components of an Ada run-time environment. Each layer in the figure is implemented by features or capabilities provided by lower layers. The Ada run-time system implements features of Ada programs that cannot be or are not placed directly in the compiled program. The target operating system, if present, provides capabilities beyond the scope of the Ada language and may also be required to assist implementation of some of the features of Ada such as tasking and input/output. The functionality of an Ada run-time environment is largely dependent on the presence or absence of an underlying target operating system. It is highly likely, for example, that stand-alone Ada run-time environments for the Digital Equipment Corporation (DEC) VAX computer will provide significantly fewer capabilities to Ada programmers than Ada run-time environments for the VAX, which permits interface to the VMS operating system. It is, therefore, important to realize that the ACPS only applies to features appearing in the Ada Language Reference Manual (ANSI/MIL-STD-1815A-1983). It includes tests of mandatory as well as compiler-dependent features (not including `LOW_LEVEL_IO`) of Ada. It does not, however, include tests of special packages provided by an Ada compiler or of capabilities provided by a target operating system. As a result, the ACPS does not address many of the capabilities of an Ada run-time environment that may be extensively used in a real-time application and that as a result can have a significant impact on performance.

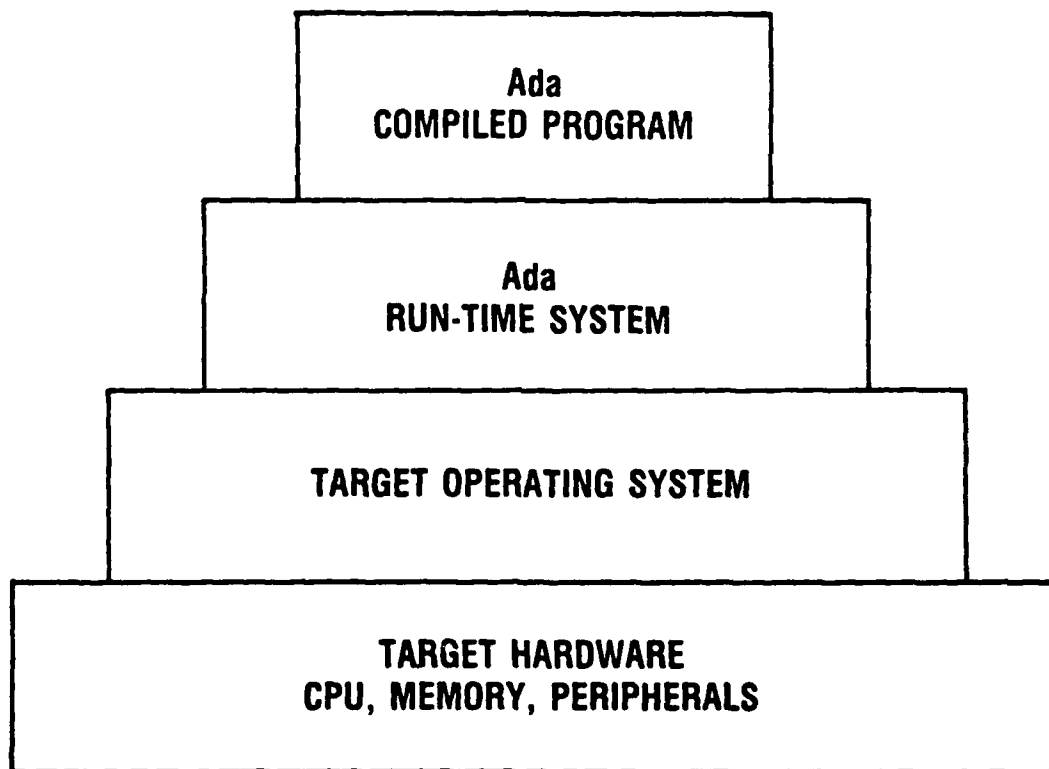


Figure 1-1. Ada Run-Time Environment Components

The forthcoming sections of this document describe the major components of the overall ACPS system and their application within the ACPS capability. Section 2 details the contents of ACPS, describing the seven types of ACPS tests and their interrelationships within the system. Section 3 discusses some of the test measurement concerns that were considered in development of the ACPS. To ensure that test results gathered and reported are accurate, repeatable, and well-focused, test measurement and coding techniques were developed and are discussed in Section 4. This section also describes sample test templates that illustrate the various forms of ACPS tests, and provides necessary data to users wishing to add more tests to the ACPS test roster. The common language test environment architecture is described in Section 5. To facilitate development of JOVIAL and FORTRAN programs from Ada programs, a set of naming conventions was devised and is detailed in Section 6. Section 7 describes the actions necessary to rehost the ACPS and also discusses the types of machine- and compiler-dependent modifications that may be necessary when rehosting the ACPS or applying the system to a different compiler. Section 8 describes methods of using the ACPS in evaluating the performance impact of entire Ada run-time environments.

The appendices are comprised of additional descriptions of key ACPS components and their functions. Appendices A and B describe the format and content of ACPS delivery tapes. Appendix C discusses methods of installing the DEC VAX/VMS version of the ACPS and provides lists of command procedures and other information useful when applying the ACPS to other compilers hosted on VAX/VMS.

To facilitate rehosting of the ACPS, almost all use of machine- and compiler-dependent language features is isolated to the test support software modules. Appendix D lists the package specifications of the Ada test support software packages. To ease comparison of test execution results, a common language report format was used and tools were developed to automatically compare the compile-time and run-time performance of Ada, JOVIAL, and FORTRAN test executions. Appendix E discusses how to use the DEC VAX/VMS version of these tools and describes with examples the format of all tool input/output files.

There are three classes of ACPS tests: (1) those unique to Ada, (2) those unique to both Ada and JOVIAL, and (3) those that can be represented in Ada, JOVIAL, and FORTRAN. To assist users in assessing the performance impact of transitioning JOVIAL and FORTRAN applications to Ada, Appendix F describes the function of each Ada test and indicates for each test whether JOVIAL and FORTRAN tests are provided. The tables in Appendix G contain Ada test program and source code file data broken down by Ada types A, C, E, S, and T. The latter tables G-6 and G-7 contain test program and source code data for JOVIAL and FORTRAN, respectively.

1.1 TEST SUITE

The core of the ACPS is the test suite. The test suite is a collection of programs each written to test a specific feature of Ada, such as the case statement or the record aggregate. For each test, the elapse time and memory usage is recorded for both the compilation and linkage stage and the execution

stage. Additionally, the support software can be customized to collect other data that is significant for the given configuration, such as the page fault rate or buffered input/output count.

To ease the conversion to new configurations, the test suite software is divided into two groups: Test Software and Test Support Software (TSS). The Test Software simply consists of the individual programs that use the language features. This group is meant to be machine and operating system-independent. The TSS is divided into the Operating System Specific Test Support Software (OTSS) and the Language Specific Test Support Software (LTSS). Figures 1-2, 1-3, and 1-4 show how the various groups merge to generate the test programs.

Five versions of the Ada test suite exist. The first is the baseline suite. The second is compiled without constraint checking, which should make the code faster and smaller. The third suite includes exception handlers in each procedure. The fourth and fifth versions are compiled with the OPTIMIZE pragma set memory optimization and speed optimization, respectively.

When taken collectively, the data for the five suites provide a thorough and detailed performance profile. This profile can be used to estimate the performance of an algorithm on a given configuration or to assist in optimizing a program. A small set of multi-featured benchmarks could not give this level of detail nor this degree of completeness. However, to allow comparison with published benchmark results, two public domain benchmarks, the Dhrystone and Whetstone tests, have been added.

The compilation and linkage data give an indication of how heavily the compiler uses the configuration's resources and the suitability of the compiler for software development. A compiler that produces exceptionally fast executable code might be too much of a burden on the system to be used constantly. In this case, a second compiler might be used during the earlier phases of development.

Since Ada is meant to replace the other languages currently in use by DoD, it was decided that the ACPS should include JOVIAL and FORTRAN versions of the test suite. JOVIAL J73 was chosen because it was the Air Force's previous standard high-order language (HOL) for embedded systems. FORTRAN, in turn, has many mature compilers that can be used for comparison of efficiency and performance.

The OTSS provides a common access path to the system data such as the system clock and the current memory usage. Its routines are shared between all of the test suites used on a particular operating system and must be provided in object code form before the tests can be run. Usually, the OTSS software will be FORTRAN or assembly language subroutines.

The LTSS is a set of procedures and data definitions that permit the test software to remain machine-independent by hiding the details of OTSS, such as what data are being collected and how the timing is being done. In addition, compiler-dependent features are isolated in the LTSS. The Ada versions of the LTSS define the default floating point and integer data types. The majority of the global variables used in the tests are defined in the LTSS files.

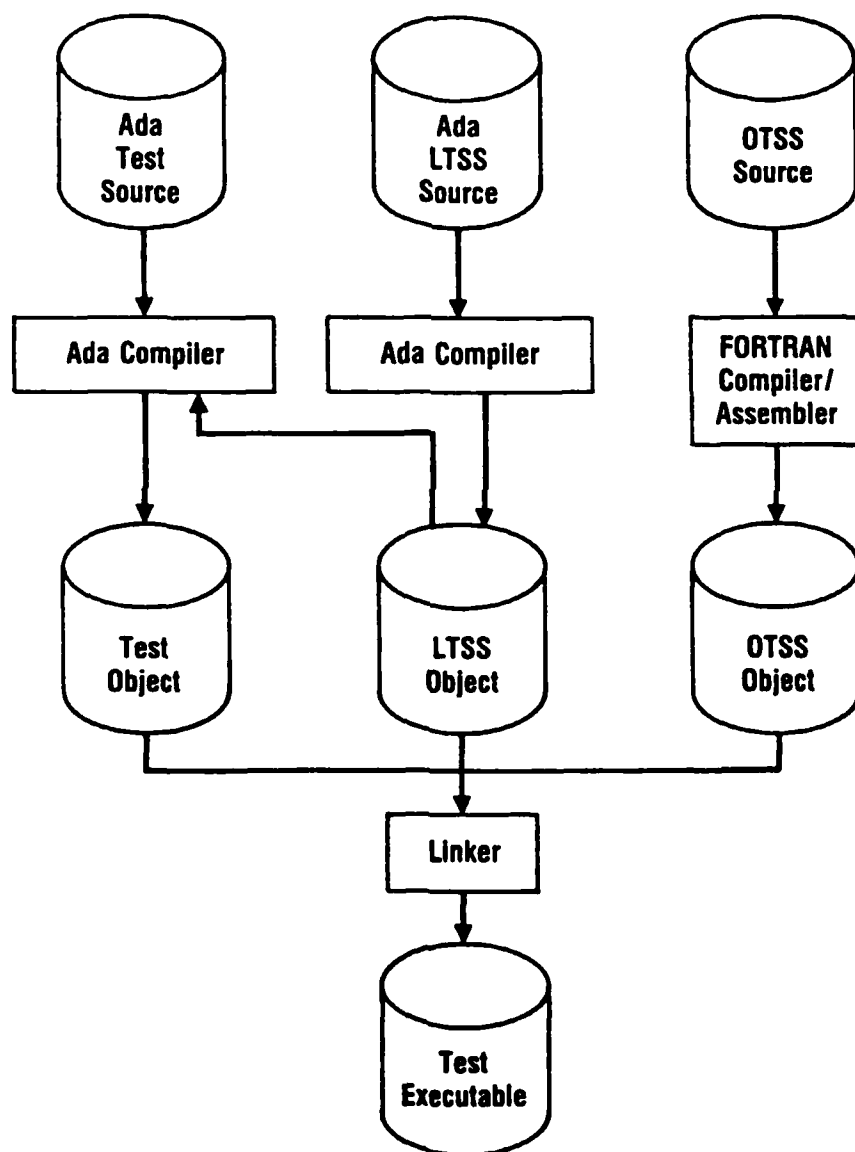


Figure 1-2. Ada Test Suite Generation

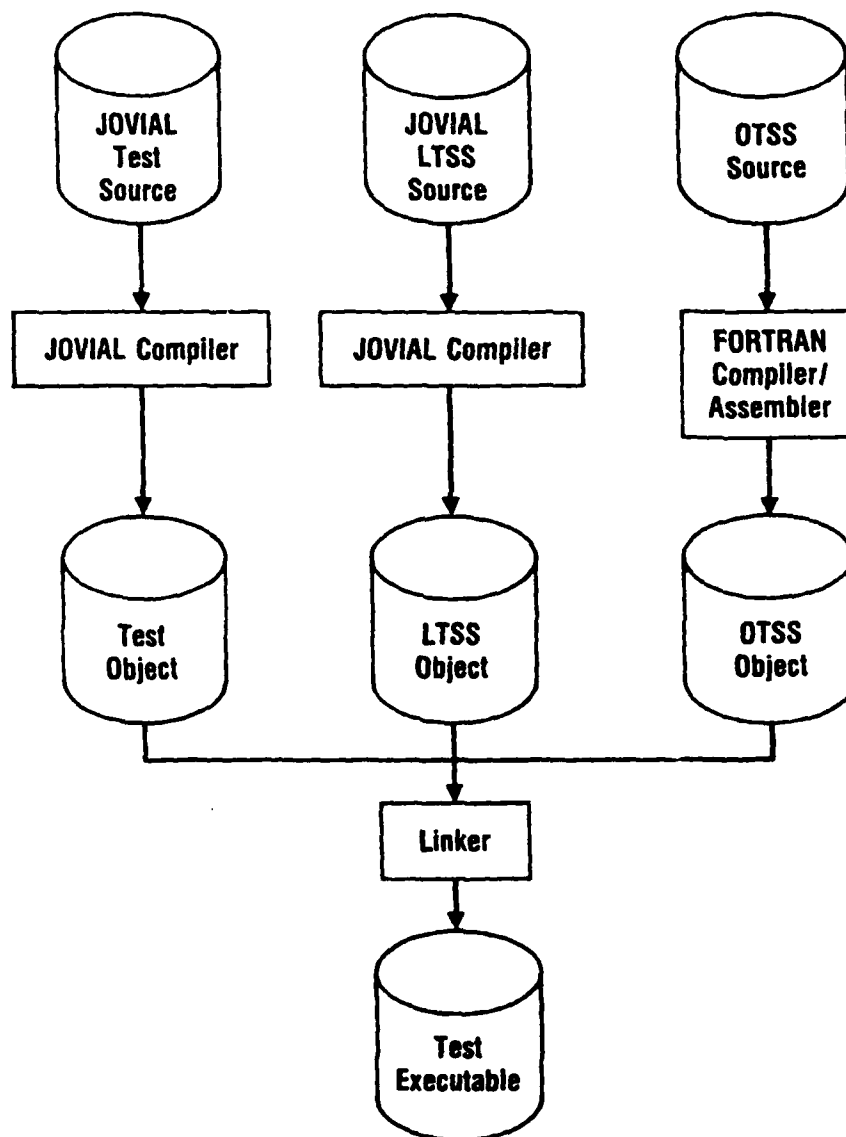


Figure 1-3. JOVIAL Test Suite Generation

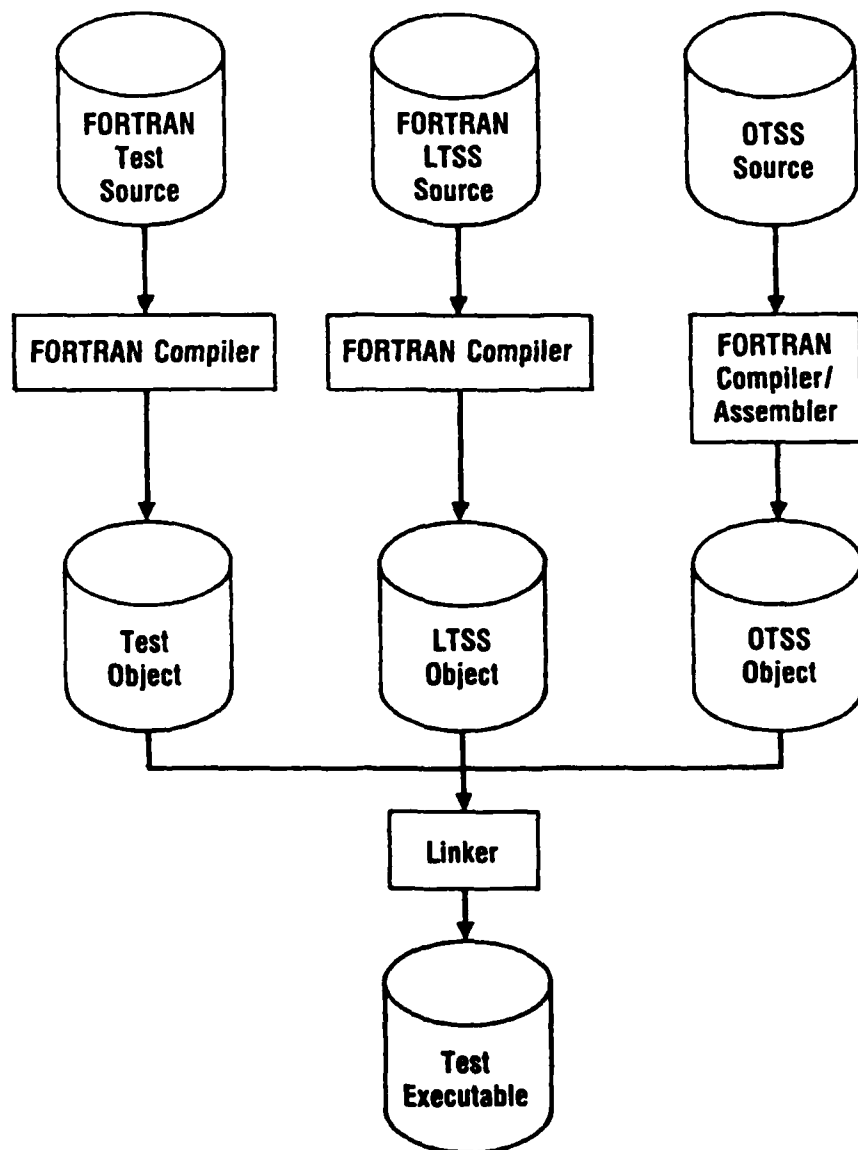


Figure 1-4. FORTRAN Test Suite Generation

1.2 TEST EVALUATION CAPABILITY

The base Ada test suite is comprised of 459 programs. Thus, it was felt that an automated method was needed to process the test suite and the resulting data. The Test Evaluation Capability (TEC) is a set of procedures written in the operating system's job control language and FORTRAN programs written to handle these functions. The TEC job control language procedures, called scripts, are divided into those that control the compilation and linkage and those that execute the test programs. The two FORTRAN programs included in the TEC take the raw data and produce formatted data files and a formatted comparison file. One program compares two compilation and linkage results, and the other program compares execution results.

2. TEST CATEGORIES

The ACPS consists of seven types of tests whose execution results are to be compared:

- Ada tests
- Ada tests with constraint checks suppressed
- Ada tests with an exception handler in every procedure
- Ada tests with space optimization pragmas
- Ada tests with time optimization pragmas
- JOVIAL J73 tests
- FORTRAN 77 tests

Almost all Ada tests use numeric data types that are defined in separate test support software packages. Two versions of these packages are supplied: one with numeric types defined with range constraints and one with numeric types defined without range constraints. All Ada tests should be executed with both packages.

Each type of test is divided into four categories: single language feature tests, multiple language feature tests, multiple task loading tests, and compiler optimization tests.

The ACPS only tests features of Ada that are described in the ANSI/MIL-STD-1815A reference manual (LRM) for Ada. The ACPS does not include tests of additional compiler-dependent pragmas or packages that are not described in the LRM. (Use of the DEC VAX Ada pragma `TIME_SLICE` and the package `MATH_LIB` are the only exceptions.) The tests are designed to measure only the performance of code generated by Ada compilers. They are not designed to measure the capacities of compilers and their associated run-time systems.

Single language feature tests, as their name implies, exercise specific language features of Ada (e.g., data type conversion). If other language features are needed in order to construct a test of this type, then an additional test is developed to account for the overhead introduced. These tests are useful in determining which features of Ada are implemented inefficiently and consequently which features of Ada could have the greatest impact on application performance.

Multiple language feature tests are designed to test a composite of language features to either perform an application-specific algorithm (e.g., Fourier transform) or to represent the frequency of language feature use for types of applications (e.g., the Whetstone and Dhrystone tests).

Multiple task loading tests are designed to stress the task scheduling, input/output and memory management capabilities of Ada implementations. Three types of tasks are used: CPU intensive, CPU and memory intensive, and input/output intensive. Task loading tests consist of a varying number of tasks of identical type (e.g., input/output intensive). They are differentiated by the type of the task and by the number of tasks executing.

Compiler optimization tests are designed to test well-known, machine-independent, code-generation optimization algorithms used in compilers (e.g., constant folding). Most of the tests are composed of two parts: a version coded to test a particular type of compiler optimization technique and a functionally equivalent hand-optimized version. Separate test results are generated for each version of the test and for the comparison of the two versions.

The ACPS is composed of single language feature tests rather than multi-language or application specific tests. This bias is due to the developer's perception that it is very difficult to draw conclusions from performance data obtained from execution of any multi-language feature tests. Unless the execution time of each statement of an application is known, one cannot understand the performance results obtained, cannot recommend changes in coding techniques to improve performance, and cannot predict the performance of other applications which use similar language features. By containing a comprehensive set of single language feature tests, the ACPS can supply sufficient detailed performance data to (a) permit informed interpretation of application test execution results and (b) provide a basis for recommendation of remedial action to the compiler vendor or to the application coder in order to improve application performance.

3. TEST MEASUREMENT CONCERNS

This section discusses the test measurement concerns that drove selection of the test measurement techniques described in Section 4 and Appendix C.

The ACPS was designed for use in evaluating the efficiency of Ada compilers for ground applications as well as spaceborne applications. The ACPS and its support software were created on a host and target machine that used a general purpose operating system (DEC VAX/VMS) which supports both real-time and nonreal-time applications. As a result, some of the test measurement techniques employed were selected due only to the fact that a general purpose target operating system would be used.

Test measurement concerns that were addressed in development of the ACPS are as follows:

a. Source of test data.

There are several aspects to test performance: test execution time (elapsed time, idle-time, run-time system overhead), memory usage (physical, virtual, compiled code, data, run-time system, page-fault rate), input/output operations (both intended and unintended) and size of input/output files. The only compiler-independent performance measurements that can be made directly within the Ada programming language are:

- Size of compiled code
- Size of data objects
- Number of input/output statements executed
- Number of times the test code is executed
- Elapsed time of a test

The other performance statistics (e.g., physical memory usage) must be obtained in a machine and run-time system-dependent manner.

b. How to take measurements.

Performance data can be obtained both from within and outside the executing program. Measurements taken outside the executing program are made before and after the program executes and include the time to load the program into memory as well as the time to remove the program from memory. For real-time applications, one is only concerned with the execution characteristics of a test, since program loading and initialization operations are performed before real-time processing begins.

There are three approaches that can be used to measure the execution time of test code from within the test program:

1. Take a single measurement before and after a test executes.
2. Enclose the test code in a loop and take measurements for each iteration through the loop and record the maximum, median, and minimum execution times.
3. Enclose the test code in a loop and take a single measurement for multiple iterations through the test code.

The first two approaches assume existence of a clock with greater resolution than the time to execute the test code. For general-purpose operating systems, the Ada clock resolution can be as high as 1 sec. For VAX/VMS, the operating system clock resolution is 10 ms. Since test code execution times were expected to be as low as 20 μ s, the third approach was used for ACPs tests.

The second approach was used in special tests (e.g., time to wake up from a DELAY statement) that cannot be timed using a single measurement for a multiple iteration test loop. For DEC VAX/VMS, this approach required development of a machine-dependent routine to compute microsecond level elapsed times. The routine used privileged code and had to be installed in the operating system.

c. How to isolate measurements to the code tested.

It is important ensure that test measurements can be isolated to the test code itself. Any overhead introduced by the measurement process should be separately measurable and taken into account. It should also be possible to separate the time to load the test code and data into memory from the time to execute the test code. To the extent possible, execution of the test measurement code should not be affected by the test code. Conversely, the test code itself should not be affected by the test measurement code. Since test code is executed within test measurement loops, one has to make sure that compiler optimization techniques do not cause removal of test code from the test measurement loops. In addition, care should be taken in constructing tests so that compiler optimization logic does not affect the intent of the test (e.g., common subexpression elimination).

d. How to determine the accuracy and repeatability of test execution time measurements.

The accuracy of test elapsed time measurements is largely determined by the resolution of the clock used (i.e., smallest nonzero difference in clock values) in relation to the duration of a test. If the clock resolution time is 10 ms and the test execution time is greater than 1 sec, then the error in the execution time measurement is less than $\pm 1\%$. There are a number of factors, however, which affect the repeatability of test measurements. These are primarily caused by

run-time system and target operating system operations which are discussed in Section 3.1. For a desired level of measurement accuracy, one must be able to measure the repeatability of test measurements and be aware of what factors affect test measurement accuracy and how they can be controlled. Separate tests are contained within the ACPS to measure clock resolution and test measurement repeatability. For some machine architectures, it may be necessary to repeatably execute every test and only use the maximum or minimum test statistics for comparison.

- e. How to ensure, to the extent possible, that consistent run-time environments for Ada, JOVIAL, and FORTRAN implementations are used for fair comparison.

Certainly, within a single machine and target operating system configuration, conventions can be established and measurement techniques employed to ensure consistent execution environments for comparison. A problem arises, however, whenever the underlying machines or target operating systems are different. For example, one may want to compare the performance of an Ada implementation on a VAX-11/785 using VMS to another Ada implementation on a VAX-8600 using VMS. It would be difficult to draw any conclusions about the comparison of test results, because it is not possible to determine to what extent the difference in performance is due to the effect of the Ada implementation and to what extent the difference can be attributed to the effect of the target machine. In situations such as these, one could use a third implementation (e.g., FORTRAN) executing on both machines to provide a reference point for the comparison.

3.1 OPERATING SYSTEM EFFECT

As mentioned above, the underlying target operating system and Ada run-time system can have a significant impact on the repeatability of test measurements. Complex operating systems such as DEC VAX/VMS can impact a test measurement in numerous ways:

- a. Other user programs on a time-shared machine can interrupt test program executions.
- b. Special network and distributed database software products can interrupt test program executions either on a time scheduled basis for status and node polling activities or on an intermittent basis dependent upon activity throughout the network.
- c. Operating system processes and routines can interrupt test programs on a scheduled or intermittent basis for accounting, time management, terminal polling, physical memory allocation adjustment, test program priority adjustment, printing, etc.
- d. Ada run-time systems can distort test measurements through task scheduling activities and unexpected storage management garbage collections.

The user typically has a good deal of control over the effect of the operating system. The tests can be run stand-alone on a computer with no other user program or batch process executing. Network, distributed database, and system processes used for accounting and printing can be disabled. However, no matter how well the ACPS user constrains the run-time environment, the ACPS test repeatability program must still be run to determine the potential variance in test measurements.

3.2 HARDWARE EFFECT

The computer system hardware can affect the test measurement and test evaluation process in two ways: It can affect the repeatability of test measurements on a single machine, and it can impact the comparisons of test executions on different machines.

The effect on test measurement repeatability will primarily be seen in input/output tests. Input/output throughput can be affected by the positioning and fragmentation of files on a single disk. The ACPS test languages do not provide mechanisms for controlling how files are stored on disks. As a result, the user may wish to run ACPS input/output tests several times and use the minimum test measurements obtained for comparison. It is also possible that hardware memory caches can also impact test repeatability. On DEC VAX, use of the high-speed memory cache is determined by physical address location. For DEC VAX/VMS, the virtual to physical address mapping will be different each time a test program is executed. For large ACPS test programs (e.g., loading tests), memory cache use could impact program performance and the user may wish to run these tests several times and use the minimum of test measurements made. Instruction pipelining can also impact test repeatability on computers for which separate memory modules can be accessed simultaneously and for which the user has no control over how his program is stored in physical memory.

The computer hardware configuration has a significant impact on evaluations of compiler implementations on different machines even for machines with the same hardware architecture. Instruction set capabilities and execution times can vary significantly. Memory access time, cache memory size and access time, disk seek and throughput times can also vary tremendously among machines supporting the same instruction set architecture. Ideally, different compiler implementations should be compared through test executions in the same hardware/software environment. If this is not practical, then a third compiler (e.g., FORTRAN) should be used on both machines to provide a point of reference for the comparison.

4. TEST MEASUREMENT AND CODING TECHNIQUES

This section discusses the techniques used to gather data that measure the performance and resource utilization of Ada, JOVIAL, and FORTRAN compilers at compile-time and of compiled ACPS tests at run-time.

Examples of the different methods used in developing ACPS tests are presented and described, which should be useful to users wishing to augment the ACPS with tests of their own.

4.1 COMPILE-TIME MEASUREMENT TECHNIQUES

Data at compile time can be derived from either the host operating system or the compiler. The ACPS compile time data processor, described in detail in Appendix E, assumes that all data are presented in a single file which consists of separate sections in the following order: compilation processing data, post-compilation processing or linkage edit processing data, source module size data, object module size data and load module size data. The expected formats for these sections, along with sample outputs, are also described in Appendix E. For the VAX/VMS version of ACPS, all data were gathered through use of operating system commands. Operating system commands were also used to display the test environment (user quotas, system activity) to ensure that the data gathered for comparison are accumulated in test environments as similar as possible.

4.2 RUN-TIME MEASUREMENT TECHNIQUES

Data are gathered at run-time by test support procedures whose Ada, JOVIAL, and FORTRAN versions are functionally equivalent in the sense that they gather the same data and report them in exactly the same format. The ACPS run-time data processor assumes that all run-time data accumulated for processing are contained on a single file in a sorted order, as described in Appendix E.

In gathering run-time test data, two problems must be addressed. The first problem is concerned with determining which data are available and how to gather them. The second problem addresses the methodology for controlling test execution such that the data gathered are accurate and measure only execution of the test code itself.

Two types of data are gathered by ACPS tests: language-specific and operating system/compiler-dependent.

Language-specific data measure the size of test code, data objects, and the elapsed time of a test. In Ada tests, the address attribute of labels is used to measure the size of test code, and the size attribute of data objects is used to compute their size. In a single test, up to five values can be output for the size of test code and data objects. Two types of elapsed time data are computed. For tests timed through repetitive execution of a test loop, the elapsed time of the test and the number of loop iterations executed are computed and output. For tests timed through single executions of test

code, multiple microsecond level measurements are made and the maximum, minimum, and median timings are computed and output. To collect language-specific data, test support procedures are used to the maximum extent possible. Their use in gathering test data is described in detail and illustrated with examples in Section 4.3.

Operating system and compiler-dependent data are additional performance measures that may be provided by an Ada run-time environment through access to special compiler-dependent run-time procedures or to accounting procedures within an underlying operating system. They may include such data as actual time spent executing machine instructions, number of input/output operations, physical memory usage, etc. The data are collected by ACPS test support software in a manner that is transparent to the test programs. Up to nine data of this type can be collected and output for processing by the ACPS run-time data comparison tool.

To ensure that the test data collected at run-time are accurate and repeatable and only measure execution of the test code itself, various coding techniques were adopted both within ACPS test programs and within the test support software.

The following coding techniques were employed to ensure to the maximum extent possible that the run-time data collected during a test apply only to the code being tested:

- a. All ACPS tests call test support software to start and stop tests and to gather and report performance data.
- b. Special tests (Table 4-1, page 4-8) were created to measure the overhead of the test support software and the test code loop. The ACPS run-time data processor uses these measurements to remove any overhead not associated with the code being tested.
- c. For some tests, the test overhead cannot be determined by the special test program in Table 4-1. For these tests, the overhead is determined directly in the program containing the test by using a special test support procedure (TCOMP). An example of this approach is discussed in Section 4.3.
- d. For nonmicrosecond level tests, two sets of data are gathered: one for the first iteration through the test loop and the second for subsequent test loop iterations. In this manner, the time to load the test code and its data into memory is isolated and removed from test measurements.
- e. All ACPS tests call a test support initialization procedure which ensures that all test support code and data are in memory before test execution begins.
- f. All ACPS tests call a test support procedure at the end of execution to output test results. This guarantees that the effect of outputting the results from one test execution does not affect execution of subsequent tests.

- g. A separate test support procedure (NOOPTM) is called within each test loop to ensure that test code is not removed from the loop by compiler optimization algorithms. The support procedure suppresses compiler optimization logic by referencing but not executing code that sets and uses all global variables used in test loops. If a test is designed to use local variables only, then these variables are by convention assigned to test support global variables within the test loop.

The following coding techniques were used to ensure the accuracy and repeatability of timing measurements:

- a. A separate test program (ADAPARM) was created to measure the resolution of the Ada clock and the representation of predefined data types. It is discussed in more detail in Section 7 and Appendix C.
- b. A separate test program (AFIRST) was created for use in measuring the repeatability of test measurements. Its use is described in more detail in Section 7 and Appendix C.
- c. The test support software outputs an error message if the test execution time is less than a predetermined value. To achieve 99% accuracy in test measurements, the test execution time should be at least 100 times greater than the resolution of the Ada clock (Sec. 7.3.2.1).
- d. The test support software does not allow a test to start until the Ada clock changes value. For microsecond level tests, this removes the effect of operating system overhead for operating systems whose overhead occurs at the change of Ada clock values. (This is true for DEC Ada on VAX/VMS.) This convention also reduces the amount of time necessary to execute tests by maximizing the accuracy of the start time of a test.
- e. To enhance repeatability of test measurements, WHILE loops rather than FOR loops are used to ensure that the loop control code does not vary with the complexity of the code tested within the loop. To control test duration, a global variable (LITERS) is used in the WHILE loop condition that is also incremented by a test support procedure (NOOPTM) within each test loop.

4.3 ACPS CODING TECHNIQUE EXAMPLES

This section describes through example several coding techniques used in developing ACPS test programs. It also shows how Ada programs are mapped into equivalent JOVIAL and FORTRAN programs.

Table 4-1 shows a line numbered example of prototype test loops for multiple iteration tests and single iteration microsecond level tests. This example can serve as a template for development of additional ACPS tests. The following description of the contents of Table 4-1 is ordered by line number:

Lines 1-13	Each ACPS test file is preceded by commentary prologue that describes the contents of the file.
Line 3	Lists the test names to which this file applies.
Line 4	Gives the name of the file.
Line 6	Lists the initials of the person(s) responsible for development and modification of the file and the date(s) the file was inserted into the ACPS or was modified.
Lines 10-12	Describe the purpose of the file.
Line 14	Allows use of test support procedures (TINIT, T2VALU, TSTART, NOOPTM, TSTOP, MSTART, MSTOP, TSTOPM, TPRINT) and test support global variables (LITERS) and constants (MICITR).
Line 15	Allows use of test support procedures (DUMPT) to print the contents of test support global variables.
Line 16	Allows use of loop iteration parameters (100000) externally defined in a package. These parameters define the duration of tests. They are grouped into packages according to test category to minimize the file editing required to change test durations.
Line 18	Notice that the main program procedure name is the same as the last six characters of the file name.
Line 20	Each ACPS test must call the procedure OURSYS.TINIT to initialize test support software.
Lines 21-28	Contain a template for development of multiple iteration tests. All test code would be inserted after line 25.
Line 21	Uses procedure OURSYS.T2VALU to compute the size of the test loop as the difference in addresses of labels which surround the test loop.

- Line 22 Uses procedure OURSYS.TSTART to start execution of the test whose name is supplied as an argument. This procedure initializes global variables that can be used by the test code. If initial values different from those set by TSTART are to be specified, then the initialization statements should be inserted between lines 20 and 22 along with the following statement:
- set_tmp: = true;
- Lines 23, 27 Define the start and end of the test loop.
- Line 24 Contains the test loop control statement. The variable LITERS is defined in the package OURSYS, initialized by TSTART to one, and incremented by NOOPTM within the loop. Notice that the loop iteration parameter is a six character name that begins with a "l" and has the same last five characters as the test name.
- Line 25 Invokes procedure OURSYS.NOOPTM for several purposes:
- a. To increment the loop control parameter LITERS
 - b. To prevent compiler optimization logic from removing test code involving calculations which use or set test support global variables
 - c. To force two sets of test results to be generated: one for the first iteration through the loop and the second for subsequent iterations through the loop
- Line 28 Uses procedure OURSYS.TSTOP to stop test execution and to compute and save all run-time test data for subsequent printing.
- Lines 33-40 Contain a template for development of microsecond level tests. The test code to be timed would be inserted before line 36 and between the calls to OURSYS.MSTART and OURSYS.MSTOP. The calls to MSTART and MSTOP needn't appear immediately within the test loop but can appear in a procedure or task invoked from within the test loop. In addition, calls to MSTART and TSTART needn't be made from the same procedure or task.
- Line 33 Invokes OURSYS.TSTART to start test execution.
- Line 34 Contains the test loop control statement. The variable LITERS is defined in the package OURSYS, initialized by procedure TSTART, and incremented by procedure MSTART. The test loop iteration parameter MICITR is defined in OURSYS as the maximum number of microsecond level measurements that can be made for a test.

- Line 35 Invokes OURSYS.MSTART to start a microsecond level test measurement.
- Line 36 Invokes OURSYS.MSTOP to compute the elapsed time of a microsecond level measurement and store the result in a test support data buffer.
- Line 38 Uses OURSYS.TSTOPM to stop a microsecond level test and compute and save all microsecond level run-time test data for subsequent printing. TSTOPM computes the maximum, median, and minimum value of all microsecond level measurements made during the test.
- Line 39 Invokes OURSYS.TPRINT to output test results previously saved by TSTOP and TSTOPM. Up to 50 test results can be saved for printing by TPRINT. If more than 50 tests are executed, then TSTOP and TSTOPM will call TPRINT whenever the data buffers become full.
- Line 40 Invokes OURDMP.DUMPT to output the contents of all global variables defined in package OURSYS. The global variable OURSYS.DEBUG controls whether printing occurs when DUMPT is called. By default, printing does not occur. To force printing, OURSYS.DEBUG can be set to boolean true in the test program or in default initialization in package OURSYS.

Table 4-2 contains an example of a template for a package defining test iteration counts. A package of this type is developed for each test category. As shown at line 16, the iteration counts are defined as constants.

Tables 4-3 and 4-4 show JOVIAL and FORTRAN versions of the Ada program in Table 4-1. These tables illustrate the following Ada coding conventions that were adopted to facilitate conversion of Ada tests to JOVIAL and FORTRAN:

- a. Only six characters are used for procedure and data object names in Ada. As a result, all procedure and data object names in Ada tests are identical to those in JOVIAL and FORTRAN.
- b. Ada, JOVIAL, and FORTRAN test names and file names only differ in the first character.
- c. Ada comments are suffixed with two dashes to facilitate translation to JOVIAL.
- d. Ada executable statements begin with at least six blanks to facilitate translation to FORTRAN.

As a result of the use of Ada coding conventions for ACPs tests, much of the work necessary to translate Ada tests to JOVIAL and FORTRAN equivalents was automated with VAX/VMS edit command procedures. The examples shown in Tables 4-1, 4-3, and 4-4 illustrate the following Ada language feature mappings to JOVIAL and FORTRAN:

- a. The use of an address attribute for a label can be implemented in JOVIAL with the LOC function and in FORTRAN with the ASSIGN statement (whose implementation is compiler-dependent).
- b. WHILE statements must be translated to DO statements in FORTRAN. Since a global variable is used as the loop control variable and is referenced by the routine NOOPTM, FORTRAN compilers must generate code to store into the variable for each iteration through the test loop. However, FORTRAN compilers may generate different loop control compiled code depending on the complexity of the test statements which can result in test measurements that are too high.
- c. WITH statements are translated to COMPOOL statements in JOVIAL and to VAX FORTRAN INCLUDE statements in FORTRAN.

As discussed in Section 4.2, for some tests, the test loop overhead cannot be determined by the special test program shown in Table 4-1. For these tests, the overhead is accounted for directly in the program containing the test. This kind of test program is composed of three types of packages: a test overhead package, one or more test code packages, and a test driver package. Examples of these package types are shown in Tables 4-5, 4-6, and 4-7.

Table 4-5 shows a package containing a single procedure that measures the test loop overhead for the IF statement test contained in Table 4-6. This procedure contains all statements (both executed and not executed) that are not part of the test but are necessary for its construction. It must be the first procedure executed in the test program. The following description of the features of the test overhead package shown in Table 4-5 is ordered by line number:

- | | |
|--------------|---|
| Line 29 | The procedure OURSYS.TINIT is called to initialize test support software variables and ensure that the support software procedures are read into memory. |
| Lines 32, 61 | OURSYS.TSTART and OURSYS.TSTOP are called to gather data from execution of the overhead test and to save the results for subsequent use and printing. |
| Line 34 | The loop iteration count (105305) used in the overhead test should have the same value as the loop iteration counts used in test code packages. |
| Line 65 | The procedure OURSYS.TSAVE must be called to identify the previous test executed as an overhead test so that run-time data can be saved for use in subsequent tests executed within the same Ada program to account for the test loop overhead. |

Table 4-6 shows an example of a test code package in which the test loop overhead has been previously computed and saved through invocation of OURSYS.TSAVE. The test support procedure OURSYS.TCOMP is called (line 27) to indicate that previously saved test loop overhead data are to be subtracted from the next test executed. Two sets of test results are generated by the procedure OURSYS.TSTOP: one including test loop overhead (AF05306) followed by one without test loop overhead (AF55306). The test name supplied to OURSYS.TCOMP must contain an "S" in the third character position.

Table 4-7 shows the test driver for tests in Table 4-5 and 4-6. The driver first calls the test that computes the test loop overhead and then calls all other tests to be executed. It finally calls OURSYS.TPRINT to output the run-time data for each test.

The test program structure shown in Tables 4-5, 4-6, and 4-7 is identical to the one used in the ACPS to determine whether machine-independent optimization algorithms are employed by a compiler. In this situation, the performance of a hand optimized version of a test is executed first and is compared with the performance of the compiler optimized version of a test. The run-time data comparison is computed as a fraction $[100 * (\text{hand_optimized} / \text{compiler_optimized})]$. To identify the comparison as fractional, a "C" must be used in the third character position of the test name argument to OURSYS.TCOMP [e.g., OURSYS.TCOMP ("AOC0508")].

Table 4-1. Ada Test Loop Overhead Program

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test:      AA00000,AA00000
4  -- File:      AA00000
5  --
6  --          REK  2/1/88
7  --
8  -- Description:
9  --
10 -- XXXX This test establishes the run time system dependent statistic --
11 -- XXXX overhead of the test support software(oursys.euramp). This test --
12 -- XXXX should be the first test run in any test sequence.
13 --
14 with oursys; use oursys;
15 with euramp; use euramp;
16 with aa0p000; use aa0p000;
17
18 procedure AA00000 is
19 begin
20   oursys.tinit;
21   oursys.t2valu(top'address,bottom'address);
22   oursys.tstart("AA00000");
23   <<top>> null;
24   while liters <= 100000 loop
25     oursys.noeptm;
26   end loop;
27   <<bottom>> null;
28   oursys.tstop;
29
30 -- execute test to determine overhead in accessing microsecond level --
31 -- time. --
32
33   oursys.tstart("AA00000");
34   while liters <= micitr loop
35     oursys.mstart;
36     oursys.mstop;
37   end loop;
38   oursys.tstopm;
39   oursys.tprint;
40   euramp.dumpr;
41 end;
```

Table 4-2. Sample Test Iteration Count Package

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation  --
2  --
3  -- Test:      .....
4  -- File:      AAOP000
5  --
6  --          REK   2/1/88
7  --
8  -- Description:
9  --
10 -- XXXX this package defines test loop iteration counts for test  --
11 -- XXXX overhead tests --
12
13 with OURSYS; use OURSYS;
14 package AAOP000 is
15
16     100000 : constant our_integer := 300000;
17 end AAOP000;
```

Table 4-3. JOVIAL Test Loop Overhead Program

```

1  start
2  "  Ada Real Time/Run Time Environment Test      Aerospace Corporation  "
3  "
4  "  Test:      JA00000,JAM0000
5  "  File:      JA00000
6  "
7  "          REK   2/1/88
8  "
9  "  Description:
10 "
11 "  This test establishes the run time system dependent statistic
12 "  overhead of the test support software(oursys,ourdmp). This test
13 "  should be the first test run in any test sequence.
14 "
15 !compool ('ourjov');
16 !compool ('jovdmp');
17 !compool ('ja0p000');
18 program a00000;
19 begin
20     tinit;
21     t2valu(loc(top),loc(bottom));
22     tstart('JA00000');
23     top;
24     while liters <= 100000;
25         nooptm;
26     bottom;
27     tstop;
28
29 "  execute test to determine overhead in accessing microsecond level  "
30 "  time."
31
32     tstart('JAM0000');
33     while liters <= micitr;
34     begin
35         mstart;
36         mstop;
37     end"loop;"
38     tstopm;
39     tprint;
40     dumpt;
41 end
42 term
```

Table 4-4. FORTRAN Test Loop Overhead Program

```

1  c  Ada Real Time/Run Time Environment Test      Aerospace Corporation  c
2  c
3  c  Test:      FA00000,FAM0000                    c
4  c  File:      FA00000                            c
5  c
6  c          REK   2/1/88                          c
7  c
8  c  Description:                                   c
9  c
10 c  This test establishes the run time system dependent statistic  c
11 c  overhead of the test support software(oursys,ourdmp). This test  c
12 c  should be the first test run in any test sequence.              c
13 c
14 c          program a00000
15 c          include 'ourfor.cmn'
16 c          include 'fa0p000.cmn'
17 c          call tinit
18 c          assign 100 to tmpis1
19 c          assign 200 to tmpis2
20 c          call t2valu(tmpis1,tmpis2)
21 c          call tstart('FA00000')
22 c          100      do 10 liters=1,100000
23 c                  call nooptm
24 c          10      continue
25 c          200      continue
26 c          call tstop
27 c
28 c  execute test to determine overhead in accessing microsecond level
29 c  time.
30 c
31 c
32 c          call tstart('FAM0000')
33 c          do 299 liters =1,micitr
34 c              call mstart
35 c              call mstop
36 c          299      continue
37 c          call tstopm
38 c          call tprint
39 c          call dumpt
40 c          stop
41 c          end

```

Table 4-5. Example IF Statement Test Overhead Package

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test:      AF05305
4  -- File:      AF05305
5  --
6  --          ARA,REK    2/1/88
7  --
8  -- Description:
9  --
10 -- XXXX This package measures the test loop overhead for tests:
11 -- XXXX
12 -- XXXX af05306
13 -- XXXX af05307
14 --
15 --          run statistic #1 = size of the test loop code
16 --
17 with oursys; use oursys;
18 with AF0P000; use AF0P000;
19 with ourdmp; use ourdmp;
20 package AF05305 is
21     procedure F05305;
22 end AF05305;
23 package body AF05305 is
24     procedure F05305 is
25
26         begin
27
28 -- initialize for the test sequence --
29         oursys.tinit;
30
31         oursys.t2valu(top'address,bottom'address);
32         oursys.tstart("AF05305");
33 <<top>> null;
34         while liters <= 105305 loop
35             oursys.nooptm;
36             if (tmpbs) then
37                 tmpbal(1) := false;
38                 tmpbal(2) := false;
39                 tmpbal(3) := false;
40                 tmpbal(4) := false;
41                 tmpbal(5) := false;
42                 tmpbal(6) := false;
43                 tmpbal(7) := false;
44                 tmpbal(8) := false;
45                 tmpbal(9) := false;
46                 tmpbal(10) := false;
47                 tmpbs2(1) := false;
48             end if;
49             tmpbs1 := false;
50             tmpbs2 := false;
51             tmpbs3 := false;
52             tmpbs4 := false;
53             tmpbs5 := false;
54             tmpbs6 := false;
55             tmpbs7 := false;
56             tmpbs8 := false;
57             tmpbs9 := false;
58             tmpbsa := false;
59         end loop;
60 <<bottom>> null;
61         oursys.tstop;
62         ourdmp.dumpr;
63
64 -- save test results for later use as base value --
65         oursys.tsave;
66         return;
67     end;
68 end AF05305;

```

Table 4-6. Example IF Statement Test

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test:      AF05306,AFS5306                    --
4  -- File:      AF05306                            --
5  --
6  --          ARA,REK    2/1/88                    --
7  --
8  -- Description:
9  --
10 --      xxxx This test measures the execution time of
11 --      xxxx 10 if statements with the then part executed.
12 --
13 --          run statistic 01 = size of test loop code
14 --
15 --
16 with oursys; use oursys;
17 with AFOP000; use AFOP000;
18 with ourdmp; use ourdmp;
19 package AF05306 is
20     procedure F05306;
21 end AF05306;
22 package body AF05306 is
23     procedure F05306 is
24
25         begin
26
27             oursys.tcomp("AFS5306");
28             oursys.t2valu(top'address,bottom'address);
29             oursys.tatart("AF05306");
30 <<top>> null;
31             while liters <= 105306    loop
32                 oursys.nooptm;
33                 if ( tmpba2(1) ) then
34                     tmpbs1 := false;
35                 else
36                     tmpbal(1) := false;
37                 end if;
38                 if ( tmpba2(2) ) then
39                     tmpbs2 := false;
40                 else
41                     tmpbal(2) := false;
42                 end if;
43                 if ( tmpba2(3) ) then
44                     tmpbs3 := false;
45                 else
46                     tmpbal(3) := false;
47                 end if;
48                 if ( tmpba2(4) ) then
49                     tmpbs4 := false;
50                 else
51                     tmpbal(4) := false;
52                 end if;
53                 if ( tmpba2(5) ) then
54                     tmpbs5 := false;
55                 else
56                     tmpbal(5) := false;
57                 end if;
58                 if ( tmpba2(6) ) then
59                     tmpbs6 := false;
60                 else
61                     tmpbal(6) := false;
62                 end if;
63                 if ( tmpba2(7) ) then
64                     tmpbs7 := false;
65                 else
66                     tmpbal(7) := false;
67                 end if;
68                 if ( tmpba2(8) ) then
69                     tmpbs8 := false;

```


Table 4-6. Example IF Statement Test (Continued)

```

70      else
71          tmpbal(8) := false;
72      end if;
73      if ( tmpba2(9) ) then
74          tmpbs9 := false;
75      else
76          tmpbal(9) := false;
77      end if;
78      if ( tmpba2(10) ) then
79          tmpbsa := false;
80      else
81          tmpbal(10) := false;
82      end if;
83      if ( tmpbsa ) then
84          tmpba3(1) := true;
85      end if;
86      end loop;
87      <<bottom>> null;
88      oursys.tstop;
89      ourdmp.dump;
90      return;
91  end;
92  end AF05306;

```

Table 4-7. Example IF Statement Test Driver

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test:
4  -- File:      AF05308
5  --
6  --      ARA,REK    2/1/88
7  --
8  -- Description:
9  --
10 --      **** This is the driver for tests:
11 --
12 --      af05305
13 --      af05306
14 --      af05307
15 --
16 with oursys; use oursys;
17 with AF05305; use AF05305;
18 with AF05306; use AF05306;
19 with AF05307; use AF05307;
20   procedure AF05308 is
21   begin
22
23
24 -- execute the test sequence --
25   f05305;
26   f05306;
27   F05307;
28   oursys.tprint;
29   end AF05308;

```

5. ACPS EVALUATION TESTING SOFTWARE ARCHITECTURE

The ACPS evaluation testing capability design, as depicted in Figure 5-1, is structured around the desire to test and evaluate several different Ada implementations as well as JOVIAL and FORTRAN capabilities. As such, a common language test and evaluation environment was designed to permit automated comparisons of results from Ada, JOVIAL, and FORTRAN test executions. This section describes the basic test and evaluation architecture and how the three languages fit into the overall capability.

Each language has its own environment and restrictions which must be addressed in order to get a working system. Those areas which are common across all three languages are collected into common capabilities within the test and evaluation capability. Those areas which are specific to a given language, implementation, or host operating system are identified and isolated to the maximum extent possible.

The test and evaluation capability is further divided into specific implementation areas of: compile and link operations, run-time evaluation operations, and result comparison operations.

The compile and link operations comprise the set of steps necessary to compile and link the application for each of the specific test cases and the different languages. Some of the tests are targeted for all three languages, others for Ada, and one for JOVIAL or FORTRAN, and some for Ada only. Initially, the DEC VAX/VMS operating system was chosen, so all of the current procedures are for this specific implementation environment. The basic design requires the correct and consistent usage of compilation/linking options and identification of specified system support files for running the test cases. The compile-time testing evaluation architecture is shown in Figure 5-2. Common language command procedure conventions were established to ensure that compile time statistic files generated for each language are in the same format for presentation to the test result comparator. Example command procedures are discussed in detail in Appendix C for DEC VAX/VMS. Appendix E describes the test statistic and comparison outputs in detail.

The run-time evaluation operations comprise the basis for the run-time test and evaluation capability. These operations define a common set of run-time operations and interfaces for use by the test programs during execution. These operations have been packaged together to form what is known as the OURSYS interface. This interface is constructed, to the extent possible, for Ada, JOVIAL, and FORTRAN. It basically provides a set of procedures and functions for timing and measuring a test case as well as a common set of data definitions and other utilities. Figure 5-3 graphically depicts the ACPS run-time testing evaluation architecture. Each set of language tests is supported by a common test interface (OURSYS) implemented in each language and buttressed by an inner target operating system-specific interface implemented in assembly language or FORTRAN. The common test interface generates test results in a common language format for presentation to the test comparator. Appendix E contains a detailed description of the test data and comparison output formats.

The result comparison operations comprise the set of steps necessary to interpret and present the results of the test execution. Data gathered from the compile, link, and execution phases of the tests are brought together from the various language and implementation environments for comparison and presentation. The ACPS test comparison tool is used to present formatted test results and comparisons to the user. It consists of two separate programs: the test compilation comparator (CCOMP) and the test execution comparator (CEXEC). As shown in Figures 5-4 and 5-5, each tool has a similar architecture. The tools take as input two unformatted test result files and generate formatted output for each input file as well as output that compares the results from common tests within each input file. Appendix E describes the format of all comparison tool interface files in detail.

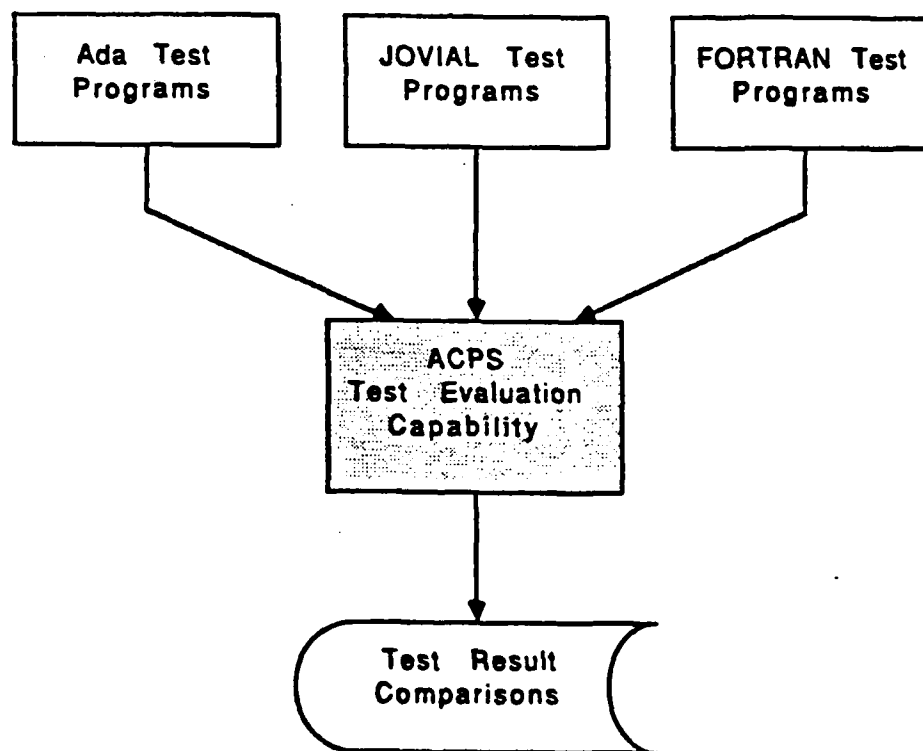


Figure 5-1. ACPS Test Evaluation Architecture



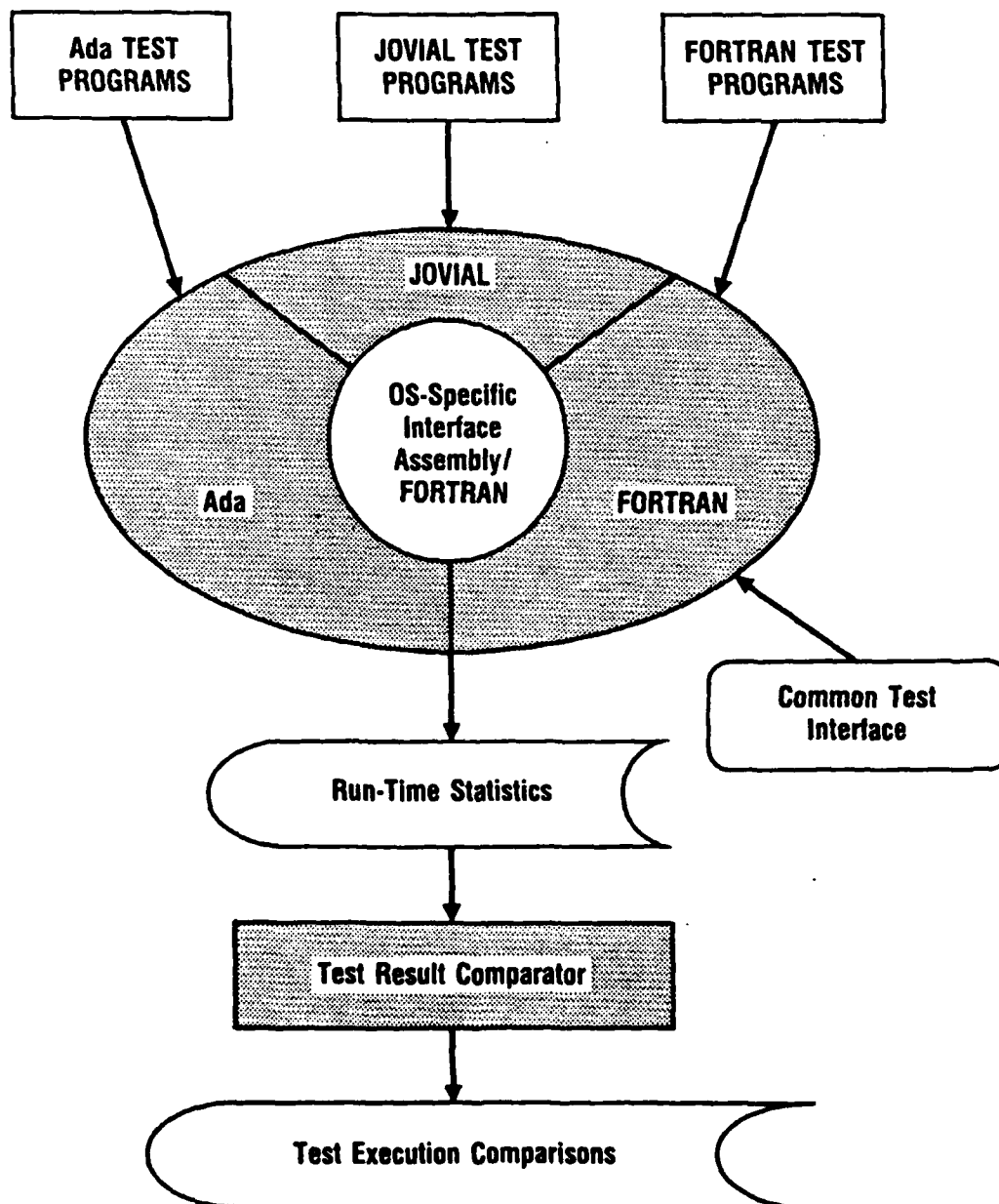


Figure 5-3. Run-Time Evaluation Architecture

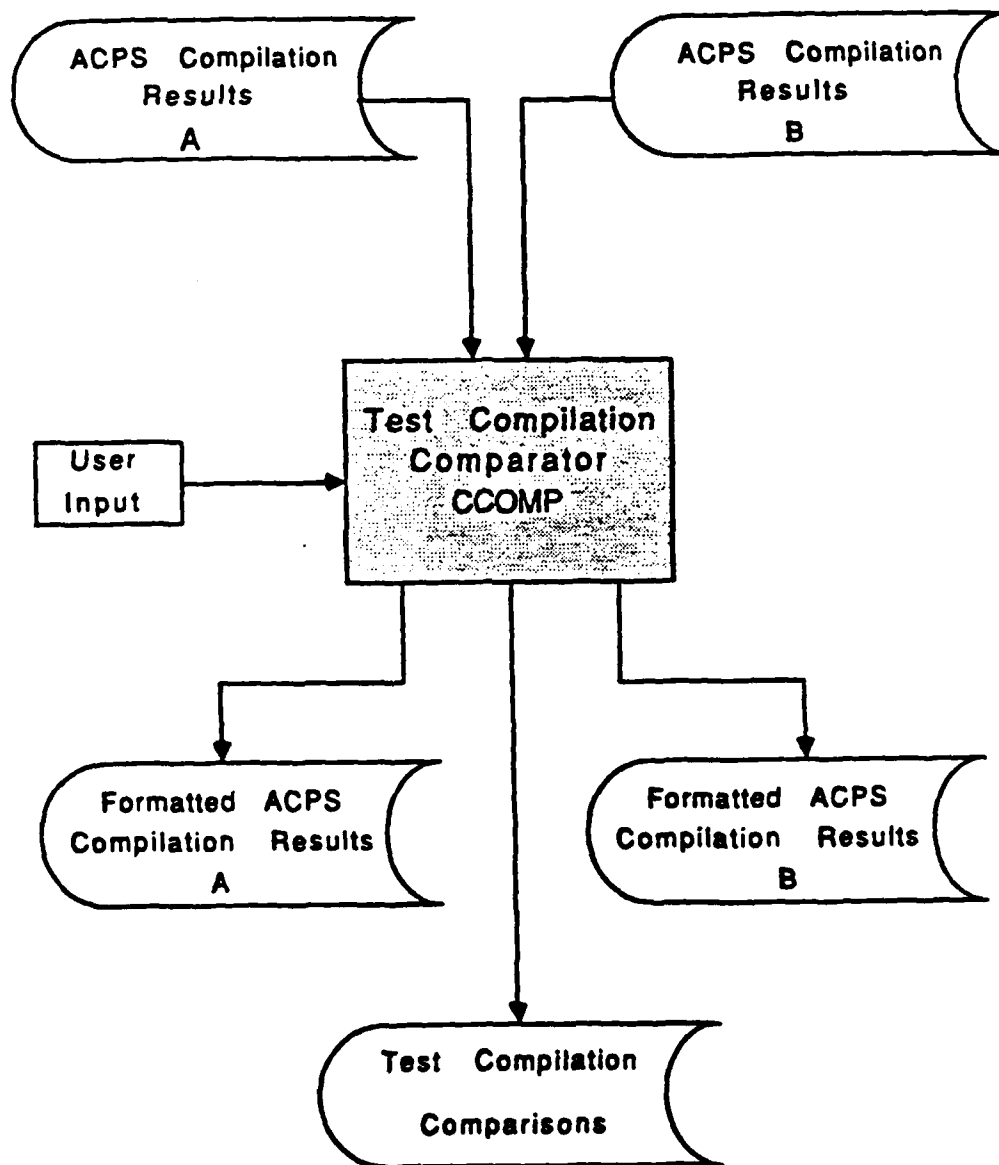


Figure 5-4. Test Compilation Comparator Interfaces

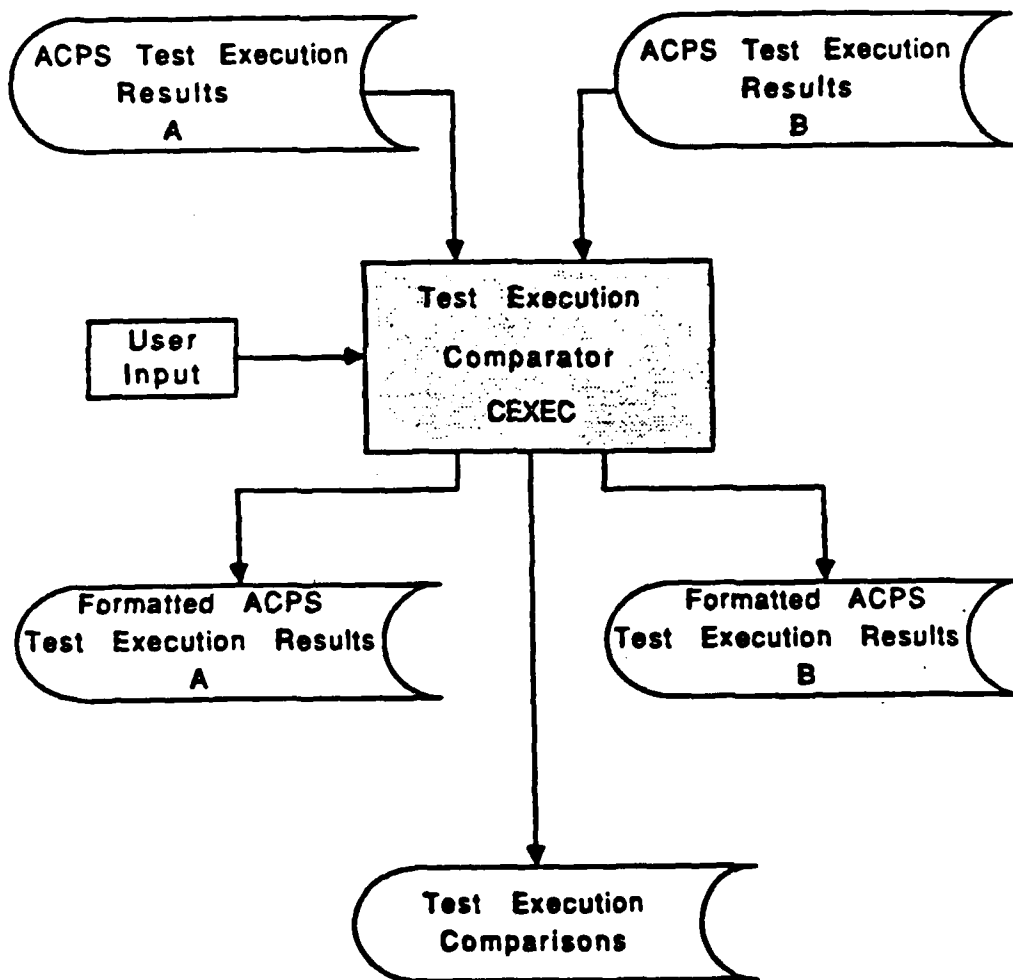


Figure 5-5. Test Execution Comparator Interfaces

6. NAMING CONVENTIONS

The primary objectives in the development of the ACPS were to ease conversion of Ada test programs to JOVIAL and FORTRAN and to minimize the effort needed to rehost the ACPS. To facilitate these objectives, three types of naming conventions were formulated: file name, test name, and programming language identifier.

File name conventions were selected so that ACPS file names would be compatible with all host computer operating systems commonly used in Department of Defense (DoD) applications. The ACPS file name syntax is identical to that of the Control Data Corporation Cyber series of computers executing the NOS BE operating system. ACPS file names are restricted to seven characters in length. The characters may be capital letters or digits. The file name must begin with a letter. Since the ACPS was developed in a VAX/VMS environment, each ACPS file also has a file type appropriate to the VMS file type conventions used by DEC VAX Ada, ECSPO JOVIAL, and DEC VAX FORTRAN compilers. Table 6-1 lists all ACPS file types. For almost all ACPS files, the file names, however, are unique and the file type field is not necessary to distinguish ACPS files within a language or between languages. However, in certain situations, multiple versions of a files may be present. The versions are distinguished by file type. For Ada test support software, file types are employed to distinguish files according to their use of optimization pragmas (SPACE, TIME). For other files, the need for additional file types arises whenever a compiler used in development of an ACPS test cannot compile single or multiple statements within the file. This can occur for statements which (a) are too complicated, (b) that invoke noncompilable procedures, or (c) that test capabilities not supported by the compiler, but which may be provided by other host environments or other compilers. In these situations, the file type USE is used for the file that includes the noncompilable statements, and compiler-specific file type (e.g., JOV) is used for the file with noncompilable statements either deleted or transferred into comments.

The conventions and formats for ACPS test names are identical to those for file names and as a result, the correspondence between test names and file names is predetermined by the naming conventions. The format of ACPS test names is similar to the format of test names used in the Ada Compiler Validation Capability (ACVC) test suite. ACPS test names are divided into fields whose contents describe much of what must be known about a test: the language the test is written in; the category of the test; which language feature is being tested; whether the test is compiler-dependent, etc. The detailed formats for ACPS file names and test names are described in Sections 6.1 and 6.2, respectively.

To facilitate conversion of Ada test programs to JOVIAL and FORTRAN, Ada identifier conventions were established. Ada identifiers in newly developed tests (i.e., tests not acquired from other sources) were restricted to conform to FORTRAN 77 syntax: containing only six or less characters; containing only letters and digits; and beginning with a letter. Therefore, all data object and procedure names in corresponding Ada, JOVIAL, and FORTRAN tests are identical.

6.1 FILE NAME FORMAT

ACPS file names are comprised of seven characters divided into five fields. The name description following is ordered by fields according to their character positions:

Character 1: indicates the type of file as follows:

- A - an Ada source file
- C - an Ada source file with constraint checks suppressed
- E - an Ada source file with exception handlers in every procedure
- F - a FORTRAN source file
- I - an input data file
- J - a JOVIAL source file
- M - a machine language source file (not currently used)
- O - an output data file
- S - an Ada source file with a space optimization pragma
- T - an Ada source file with a time optimization pragma

Character 2: indicates the test category as follows:

- A - test loop overhead test
- F - language feature test
- G - general or application specific test involving multiple language features
- L - multi-task loading test either CPU or input/output intensive
- O - compiler optimization test

Character 3: dependent on test category and indicates the following:

- D - the file is involved in a test of compiler-dependent language features
- F - the file contains FORTRAN modules used by JOVIAL test programs
- J - the file contains a JOVIAL compool or FORTRAN common block definition
- M - the file contains a microsecond level language feature test
- N - the file contains a microsecond level language feature test involving compiler-dependent language features
- zero - has no special significance

Character 4: dependent on test category as follows:

- zero - for test categories A, G, and O

LRM Chapter - for test categories F and L. The character is in hexadecimal notation and indicates that the file is involved in a test that applies to this chapter in the Ada language standard (MIL-STD 1815A).

P - for all test categories. The character indicates that the file contains test loop iteration parameters.

Characters 5-7: used as test number identification. The characters may be numeric or alphabetic.

6.2 TEST NAME FORMAT

ACPS test names contain seven characters divided into five fields. The name description that follows is ordered by fields according to their character positions:

Character 1: indicates the type of test as follows:

- A - an Ada test
- C - an Ada test with constraint checks suppressed
- E - an Ada test with exception handlers in every procedure
- F - a FORTRAN test
- J - a JOVIAL test
- M - a machine language test
- S - an Ada test with space optimization pragmas
- T - an Ada test with time optimization pragmas

Character 2: indicates the test category as follows:

- A - a test loop overhead test
- F - a language feature test
- G - a general or application specific test involving multiple language features
- L - a multi-task loading test that is either CPU or input/output intensive
- O - a compiler optimization test

Character 3: dependent on test category and indicates the following:

- C - an optimization comparison test
- D - a test involving compiler-dependent language features
- I - an information test. The run statistics fields contain the test information
- M - a microsecond level language feature test
- N - a microsecond level language feature test involving compiler-dependent language features
- S - a language feature test with test loop overhead statistics removed
- zero - has no special significance

Character 4: dependent on test category as follows:

- zero - for test categories A, G, and O
- LRM Chapter - for test categories F and L. The character is in hexadecimal notation and indicates that the test applies to this chapter in the Ada language standard (MIL-STD 1815A).
- Characters 5-7: used as test number identification. These characters containing either a capital letter or a numeric digit. For many language features tests, character 5 indicates that the test applies to this section of the LRM chapter specified by character 4.

Table 6-1. ACPS File Types

File Type	Purpose
Ada	Used for Ada source code modules. Each file contains either a main program, a package, a package specification, or a package body.
CMN	Used for FORTRAN common definition modules that are referenced by machine-specific INCLUDE statements in FORTRAN source code files. Each file corresponds to an Ada package specification.
CPL	Used for JOVIAL compool definition modules. Each file corresponds to an Ada package specification.
DAT	Used for ACPS test program input/output files and for files referenced by JOVIAL !COPY directives.
DEC	Used for Ada source code modules that contain the DEC VAX Ada-specific language pragmas VOLATILE and TIME_SLICE.
FOR	Used for FORTRAN source code modules. Each file is either a main program, BLOCK DATA program, or collection of FORTRAN routines.
JOV	Used for JOVIAL source code modules. Each file corresponds to an Ada main program or an Ada package body.
SPA	Used for Ada test support software modules that contain the pragma OPTIMIZE(SPACE).
TIM	Used for Ada test support software modules that contain the pragma OPTIMIZE(TIME).
USE	Used for Ada, JOVIAL, or FORTRAN source code modules that were not used on DEC VAX/VMS because either the files were not compilable or executable or because an equivalent VAX version of the file was used. The FORTRAN files with this type may contain common definition modules, and the JOVIAL files may contain compool definition modules.
VAX	Used for modules that test VAX-specific language features. These files are not placed on the ANSI tape.
VAXSPA	Used for test support software modules that contain VAX-specific language features and the pragma OPTIMIZE(SPACE). These files are not placed on the ANSI tape.
VAXTIM	Used for test support software modules that contain VAX-specific language features and the pragma OPTIMIZE(TIME). These files are not placed on the ANSI tape.

7. ACPS INSTALLATION AND TEST EXECUTION

This section discusses installation and execution of the various parts of the test and evaluation capability. As described earlier, the testing process involves three basic phases: compilation, execution, and comparison. Each phase involves a number of different aspects about the test capability depending on the language(s) and environment being addressed. For the initial release of the test capability, the DEC VAX/VMS operating system is the execution environment used.

The compilation and execution phases are controlled by command procedures which invoke the specific compiler(s) for the specified test cases, link the results together, and invoke execution of the test cases. Results are then processed by the comparison programs for presentation to the user. For DEC VAX/VMS, these command procedures and comparator programs are already provided. For other operating systems, the user will need to construct similar operations to provide the same functionality. In addition, some low-level operating system interface operations will have to be supplied in order to access operating system dependent statistics.

The following subsections describe the tasks necessary to install, compile, and execute the ACPS tests and to use the test comparison programs. The discussion will be as general as possible and will be oriented toward users needing to execute the ACPS tests on operating systems other than DEC VAX/VMS. References will be given as appropriate to appendices for detailed descriptions and examples of outputs. Appendix C describes what must be done to use the ACPS test and evaluation capability for DEC VAX/VMS.

7.1 INSTALLATION OF ACPS SOURCE FILES

The ACPS source files are delivered on an ANSI formatted tape generated on a DEC VAX/VMS system using the COPY command. The tape format and content are described in detail in Appendix A. The procedures to read the tape are host machine dependent. The tape contains four types of files: Ada test files, JOVIAL test files, FORTRAN test files, and comparison tool source files.

All ACPS test files were developed to compile and execute properly on the DEC VAX/VMS 4.X operating system using the DEC VAX Ada V1.4-33 compiler, the ECSP0 JOVIAL FV-03.000 compiler hosted and targeted to VAX/VMS, and the DEC VAX FORTRAN V4.7-271 compiler. The language-specific test support software references external procedures that gather target operating system dependent performance statistics. These procedures as well as command procedures to compile and execute ACPS programs are not provided on the ANSI delivery tape.

As discussed in Section 6.1, the file type USE for a source file indicates that the file was not used in the DEC VAX/VMS version of the ACPS but should be used when rehosting the ACPS. In some situations, another version of the file (e.g., with file type ADA) is provided on the ANSI tape. This version of the file was used on DEC VAX/VMS because the USE version wasn't compilable or executable. Tables G-1 to G-7 in Appendix G list all test program files contained on the ANSI tape and indicate which files are supplied with multiple versions.

Subsequent subsections discuss the machine-dependent modifications that must be made to the supplied files and the additional software and command procedures that must be created to fully utilize the ACPS test and evaluation capability.

7.2 CREATION OF COMMAND PROCEDURES

Test compilation and execution command procedures are not supplied with the ACPS. Tables G-1 to G-7 in Appendix G list the programs and supporting files for all ACPS test types and test languages. The file names of input/output files for all ACPS tests are constructed by the language-specific test support procedures FINAME and FIFORM. Table 7-1 lists the file names of input/output files used by all type A(Ada) tests. The same file names are used by corresponding tests in other test types (types C, E, F, J, S, and T). Table 7-2 lists the FORTRAN unit numbers assigned to files in input/output tests. All Ada tests write test result data output to the default output device. All JOVIAL and FORTRAN tests write test data output to unit 6, which is not connected to an external file.

ACPS test comparison programs use up to eight input/output files. Appendix E describes the purpose, format, and FORTRAN unit number of each comparison program input/output file. FORTRAN OPEN statements are only used to assert whether the file is formatted or not and whether it is to be read or written. OPEN statements are not used for terminal input/output units.

If the ACPS test comparison programs are to be used, compilation and execution command procedures must be constructed so that the test result files generated can be processed by the comparison programs. Appendix E describes the content and format expected by the comparison programs for compile-time and run-time test result files with operating system-dependent records and data specific to DEC VAX/VMS. The comparison programs not only expect input test result files to be in a specific format; they also expect file compilations, program linkage edits, file size computations, and test executions to appear in a specific order. This ordering is defined in Section E.3. Tables G-1 to G-7 in Appendix G show the order that programs should be compiled and executed.

There are seven types of ACPS tests supplied. Separate compilation and execution command procedures should be created for each type of test in order to perform comparisons (e.g., type A versus type J, type A versus type C, etc.). There are two versions of each of the packages OURSYS and OURTYP: one with range constraints on numeric types (files OURSYSR and OURTYPR) and one without (files OURSYS and OURTYP); all Ada tests should be compiled and executed with both versions. Several of the Ada test support software files are supplied with the file types SPA and TIM. These files should be used for the S and T type Ada tests, respectively. Table 7-3 lists the file types of all language-specific test support software files supplied on the ANSI tape.

7.3 MACHINE-DEPENDENT SOFTWARE MODIFICATIONS

This section discusses the machine- and compiler-dependent language features that are used in ACPS source code files and also the kinds of source code modifications that may need to be made when rehosting the ACPS. The ACPS was designed so that almost all statements using machine-dependent language features are placed in the test support software. If a particular feature used in a test support package is not supported, Tables 7-1 through 7-4, can be used to assist identification of the programs which must be deleted, since these figures list which test support packages are used by each test program.

7.3.1 Machine-Specific Test Support Software

Several target machine and target operating system-dependent external procedures are called by language-specific test support software to access microsecond level time and run-time environment performance data. These procedures are not supplied with the ACPS and must be developed. A description of the purpose of and arguments to these procedures is contained in Appendix D. All JOVIAL and FORTRAN tests assume that the function SECS (described in Appendix D) will be supplied to permit test support software to compute elapsed execution times. If the capability to access microsecond level time cannot be provided, then none of the microsecond level tests (i.e., tests names with M or N in character position three) can be executed. If additional run-time environment data can be supplied, then modifications are required to the run-time test comparator to process and output these data.

7.3.2 Language-Specific Test Support Software

The language-specific test support software is used by ACPS tests to control the gathering and reporting of test result data. Functionally equivalent capabilities are provided for each test language. Table 7-4 describes all language-specific test support software files and shows the correspondence of files among the three test languages. Compiler- and machine-dependent statements are identified in the Ada test support software by special comment strings (--*) placed at the end of the statement. The user must manually identify the corresponding statements in the functionally equivalent JOVIAL and FORTRAN files. The modifications that may need to be made to the test support software are as follows:

- a. Interface to machine-specific test support software. If machine-specific test support software (Sec. 7.3.1) cannot be developed or an Ada compiler does not permit interface to them, then stubs for each of the requisite procedures should be developed and integrated into the test support software.
- b. ACPS input/output. File name dependencies are encapsulated in the procedure FINAME. Ada FORM parameter dependencies are handled by the procedure FIFORM. If a FORTRAN compiler is not available, then the FORTRAN routines in files SYSJOV and TMPDMP that perform input/output for JOVIAL test support software must be redeveloped. Ada test support software uses the predefined Ada generic package TEXT_IO for reporting test results. If this package is not supported, then Ada statements that output test results must be changed to invoke equivalent functions in another language.

- c. Numeric data types. ACPS tests were developed on the DEC VAX computer which has numerous machine level integer and floating point data types. Section C.5 discusses the mapping between ACPS numeric types and Ada predefined types. (Note: The ACPS type OUR_D_FLOAT is only used for the DEC VAX/VMS version of the ACPS.) The range constraints supplied for these data types in packages OURSYSR and OURTYPR are based on the corresponding size of the underlying VAX data type (as shown in Section C.5) and may need to be changed. The FORTRAN version of OURTYP contains nonstandard FORTRAN syntax for data type specification statements. These statements may have to be modified for each machine-specific FORTRAN compiler. Some of the ACPS data types (e.g., OUR_SHORT_SHORT_INTEGER) may not be supported by each language specific compiler. This may be resolved by either changing the machine data type mappings, or by modifying or deleting the ACPS tests that use the unsupported data types. However, in order to obtain meaningful comparisons between ACPS test execution for different compilers on the same machine, the underlying machine data type for each ACPS data type should be the same for all compilers used.
- d. Representation specifications. The bit level storage allocation statements in package OURSPC are machine and compiler dependent and may need to be changed, or the associated ACPS tests may have to be deleted or modified.
- e. Pragma PACK. Support of the PACK pragma is compiler-dependent. It is used by package OURSPC. Its use may have to be deleted in OURSPC and the associated ACPS tests modified or deleted as well.
- f. Address arithmetic. The procedure T2VALU is used to compute the difference in address values of its arguments. This computation is compiler-dependent and may have to be deleted. Since every Ada test program references address attributes of labels, many files may need to be modified. One approach that may be used is to declare objects (e.g., top, bottom) in package OURSYS that have the same name as program labels. For a given machine, the address value calculation should be expressed in the same units for all compilers. For DEC VAX, results from address calculations are expressed in units of bytes.
- g. Elapsed time calculations. The accuracy of elapsed time calculations should be consistent across compilers. JOVIAL and FORTRAN test support software reference the machine-specific function SECS, which returns the current value of time as a floating point value in units of seconds. If the resolution of the Ada clock is worse than the resolution of values returned by SECS, then SECS should be used in Ada support software to access time. SECS will be invoked by Ada support software if the value OS is assigned to the variable TCLOCK in the file ADASYS.

- h. Test loop overhead statistics. The test execution comparison program is intended for use in accounting for the overhead introduced by the test support software in controlling test executions. If this tool cannot be used (e.g., on machines that do not support FORTRAN); then variables in the test support software must be set to allow the test support software itself to account for the overhead it introduces into test executions. A special program in each test language (AFIRST, FFIRST, JFIRST) is used to measure the values to be assigned to these variables. A more detailed description of these variables and the test programs used to calculate their values is contained in Section C.7.
- i. FORTTRAN global common reference. In FORTRAN test support software, Ada package specifications are implemented in separate nonexecutable source files. These source files are then referenced by VAX FORTRAN INCLUDE statements where necessary. These INCLUDE statements will need to be modified for each FORTRAN compiler used.
- j. Multiple FORTRAN program units per file. Several FORTRAN test support software files contain more than one program unit or subroutine. These files will need to be partitioned into separate files for compilers that allow only one program unit per file.
- k. Math function names. The Whetstone test exercises numerous mathematical functions while other tests exercise the same functions individually. The package MATHFUN is used to make these functions available for ACPS data types with the same names used in the Whetstone test. This package is compiler-dependent and may need to be modified. If the run-time environment does not support access to math functions, then all tests using them should be deleted.

7.3.2.1 Determination of Ada Compiler Dependent Parameters. The program ADAPARM is used to compute the resolution of the Ada clock, the minimum delay interval in a DELAY statement, the size of predefined data types, and the name of the standard output file. Section C.5 discusses the use of this program and shows sample output from it for DEC VAX Ada.

7.3.3 ACPS Tests

Many ACPS tests involve the use of compiler-dependent features of Ada which may not be supported or which may require source language changes in order to execute. All ACPS files which are involved in tests of Ada language features that are known to be compiler-dependent contain the symbol D or N in the third character position of the file name (Sec. 6.1). Any statements in an Ada source file directly using compiler-dependent features that may require modification will be identified with a comment appearing at the end of the statement (--*). Location of the corresponding statements in the equivalent FORTRAN and JOVIAL source files must be determined manually. Tables G-1 through G-5 in Appendix G underline all Ada test files that may require source code modifications to compile or execute properly. Other machine-dependent test files use features that may not be supported (e.g., input/output) and

that cannot be made to work through modification of the source code. For these files, to the extent possible, the use of compiler-dependent features has been placed in the test support software. Compiler-dependent language features used include the following:

- input/output statements
- pragmas `INLINE`, `OPTIMIZE`, `PACK`, `PRIORITY`, `SHARED`, `SUPPRESS`
- task scheduling algorithms
- use of a special pragma (`TIME_SLICE` for DEC VAX Ada) to force timed sliced scheduling of tasks of equal priority
- use of predefined numeric data types other than integer and float
- address and `storage_size` attributes
- representation specifications
- math function names
- change of representation
- nonstandard data type conversions
- unchecked storage deallocation

All values computed in integer calculations are less than 16 bits in magnitude. Test iteration counts can be greater than 2^{16} and may need to be modified for execution on 16-bit machines.

All tasking tests that use the priority pragma assume that three priority levels are available and that a preemptive task scheduling algorithm is used by the Ada run-time system. Many tests use a `DELAY` statement in a high priority task to allow lower priority task(s) to be executed and the assumption is made that at the end of the delay interval the high priority task will preempt any lower priority tasks from executing. In addition, task loading tests assume that task execution can be completed within `OURSPC.max_time` seconds. If a task in a type A multi-tasking test aborts, the user will not be notified as to why the task aborted. The user should therefore execute the type E version of all Ada tasking tests first. Since these tests contain exception handlers in all procedures, the user will be notified about the type of any error that occurs for corrective action (e.g., increasing the `storage_size` attribute for a task).

For JOVIAL tests, FORTRAN subroutines are used for input and output operations. If a FORTRAN compiler is not available, then the FORTRAN subroutines in files `SYSJOV` and `TMPDMP` will have to be redeveloped.

For FORTRAN tests, all executable source files use VAX FORTRAN INCLUDE statements to reference global common definitions. These statements will have to be modified for each FORTRAN compiler used. In addition, some test files contain more than one program unit or subroutine. These files will need to be divided into separate files for compilers that allow only one program unit per source code file. For these compilers, all local procedure tests (see Appendix F) can be deleted.

7.3.4 ACPS Test Comparison Tool

The ACPS test comparison tool consists of two programs: a compile-time test result comparator and a run-time test result comparator. Both programs are implemented in FORTRAN 77 and utilize the VAX FORTRAN INCLUDE statement to import global common specifications. The programs are divided into operating system-dependent modules and operating system-independent modules. The target operating system-dependent modules for VAX/VMS are included and identified in the source files and must be redeveloped for each new host and target operating system. The comparison tool consists of the following source files: CCOMP, CCOMPA, CCOMPV, CEEXEC, CEEXECA, CEEXECV. The files CCOMPV and CEEXECV are VAX/VMS-specific and are provided to ease rehosting of the comparison tool.

7.4 DETERMINATION OF TEST DURATION

Test durations are determined by test iteration counts which are grouped into files according to test category (Sec. 6.1). The iteration counts were set so that most type A Ada tests execute for at least 1 sec exclusive of test overhead using DEC VAX Ada on a VAX-11/8600.

The duration of a test is essentially driven by the desired accuracy of test measurements. For a desired measurement accuracy of 99%, tests (exclusive of loop overhead) must be executed for a duration at least 100 times greater than the resolution of the clock used. In addition, this measurement accuracy must be achieved for every test execution. As discussed in Section 3, the repeatability of elapsed time measurements is impacted by the run-time system architecture. A separate test program, AFIRST, is used to measure the repeatability of test measurements. The use of AFIRST in demonstrating test measurement accuracy for DEC VAX/VMS is discussed in Section C.8. AFIRST should be executed at least 50 times so that approximately 1000 test measurements are made for the same test code. The variance in test measurements made determine the length of time required to run a test to achieve an expected level of test measurement accuracy. The test iteration counts can then be adjusted accordingly.

7.5 DETERMINATION OF TEST MEASUREMENT ACCURACY

Section 7.4 discusses varying the test durations of a single program (AFIRST) to ensure test repeatability. Although this process does account for inaccuracies in the clock used and for scheduled or intermittent interference from the run-time system or underlying operating system, it does not account for all factors that impact program performance for all machine architectures. For the DEC VAX computer, it has been seen that the amount of physical memory

can have a significant impact on the execution efficiency of test programs. Single test program execution speeds have varied by more than 30% irrespective of test duration for large memory configurations (e.g., 48 Mbyte). It is believed that this variance is due to the combined effect of instruction pipelining and how programs are mapped to physical memory. For some machine architectures, therefore, it is necessary to compute the variability of each test measurement made.

As shown in Sections C.10 and E.24, the run-time test comparator tool can be used to determine the accuracy of test measurements by computing the maximum and minimum values of test execution data for each test which is repeatedly executed. The test should be repeatedly run until the maximum and minimum run data do not change. Only the maximum or minimum data for each test should be used for comparison.

7.6 PERFORMANCE OF TEST COMPARISONS

The ACPS was designed to encourage the user to perform as many test execution comparisons as possible. For a single Ada compiler, over 10,000 test comparisons can be made among ACPS test programs processed by that compiler. To fully utilize the capabilities of the ACPS, it is imperative that the ACPS test comparison tools are installed and used.

The ACPS consists of seven types of benchmark test programs as described in Section 2. The ACPS test evaluation capability was designed for comparison of sets of tests of each type. Appendix F describes each individual test for each test type. These test descriptions also suggest additional comparisons (e.g., see the description for test AFD4135) that can be made by the user but currently not automatically by the test execution comparison tool. Five sets of Ada tests are provided. Five additional sets of executable Ada tests should be created by compiling each Ada test program using two versions of the Ada packages OURSYS (files OURSYS, OURSYSR) and OURTYP (files OURTYP and OURTYPR). Compiler options can also be used to increase the number of comparisons that can be made. For example, DEC VAX Ada permits suppression of compiler optimization algorithms. By using this, option, ten additional sets of Ada tests can be generated. Also, DEC VAX FORTRAN not only allows suppression of optimization algorithms but also permits inclusion of run-time array bound checks. For DEC VAX FORTRAN, one could therefore easily create four different test sets from a single set of test programs.

When comparing the performance of two different Ada compilers, ten different comparison options are available (comparison of type A, C, E, S, and T tests which are compiled both with and without range constraints on numeric data types). For a given Ada compiler, numerous comparison options are available to determine the effect of using range constraints on numeric types, of suppressing constraint checks, of using space or time optimization pragmas, and of using exception handlers in procedures.

When comparing the performance of different compilers for a given set of tests, it is important to make sure that the machine representation of each numeric data type used is the same for both compilers and also that the size

of components of corresponding record structures is the same. ACPS language feature tests for Section 13.7 of the LRM (tests AFOD720 to AFDD72I) display the size of all numeric data types used, the size of all composite data types used and the size of all record data type components. These data are processed for comparison by the test execution comparison tool. The Ada program ADAPARM determines the internal representation of all numeric data types used in ACPS test programs, as described in Section C.5.

The most difficult and time consuming task in generating test comparisons is to ensure that each test executes for the necessary amount of time to guarantee a desired level of test measurement accuracy. In construction of some test programs, little concern was given as to how long each test would have to be executed to ensure accurate test measurements. As a result, some of the difference tests (e.g., AFS5406) may require exorbitant computer time to achieve the desired test duration. The test iteration counts supplied on the ACPS ANSI tape were generated as follows. First, the A type tests were executed using DEC VAX Ada on a VAX 8600. A software tool was then used to adjust the iteration counts based on the execution times obtained, with the constraint that no test should be required to execute for more than 30 sec. These iteration counts were then used for all of the other test types.

Table 7-1. Ada Type A Test Program Input/Output Files

<u>PROGRAM</u>	<u>INPUT/OUTPUT FILES</u>				
AFDE220	OFDE201 OFDE206 OFDE211 OFDE216	OFDE202 OFDE207 OFDE212	OFDE203 OFDE208 OFDE213	OFDE204 OFDE209 OFDE214	OFDE205 OFDE20A OFDE215
AFDE250	IFDE231 IFDE236 IFDE241 IFDE246	IFDE232 IFDE237 IFDE242	IFDE233 IFDE238 IFDE243	IFDE234 IFDE239 IFDE244	IFDE235 IFDE23A IFDE245
AFDE420	OFDE401 OFDE406 OFDE411 OFDE416	OFDE402 OFDE407 OFDE412	OFDE403 OFDE408 OFDE413	OFDE404 OFDE409 OFDE414	OFDE405 OFDE40A OFDE415
AFDE450	IFDE431 IFDE436 IFDE441 IFDE446	IFDE432 IFDE437 IFDE442	IFDE433 IFDE438 IFDE443	IFDE434 IFDE439 IFDE444	IFDE435 IFDE43A IFDE445
ALDE2C1	OLE2001				
ALDE2C2	OLE2001 OLE2006	OLE2002 OLE2007	OLE2003 OLE2008	OLE2004 OLE2009	OLE2005 OLE2010
ALDE2C3	OLE2001				
ALDE2C4	OLE2001 OLE2006	OLE2002 OLE2007	OLE2003 OLE2008	OLE2004 OLE2009	OLE2005 OLE2010
ALDE2C5	ILE2001				
ALDE2C6	ILE2001 ILE2006	ILE2002 ILE2007	ILE2003 ILE2008	ILE2004 ILE2009	ILE2005 ILE2010
ALDE2C7	ILE2001				
ALDE2C8	ILE2001 ILE2006	ILE2002 ILE2007	ILE2003 ILE2008	ILE2004 ILE2009	ILE2005 ILE2010
ALDE4C1	OLE4001				
ALDE4C2	OLE4001 OLE4006	OLE4002 OLE4007	OLE4003 OLE4008	OLE4004 OLE4009	OLE4005 OLE4010
ALDE4C3	OLE4001				
ALDE4C4	OLE4001 OLE4006	OLE4002 OLE4007	OLE4003 OLE4008	OLE4004 OLE4009	OLE4005 OLE4010
ALDE4C5	ILE4001				
ALDE4C6	ILE4001 ILE4006	ILE4002 ILE4007	ILE4003 ILE4008	ILE4004 ILE4009	ILE4005 ILE4010
ALDE4C7	ILE4001				
ALDE4C8	ILE4001 ILE4006	ILE4002 ILE4007	ILE4003 ILE4008	ILE4004 ILE4009	ILE4005 ILE4010

Table 7-2. FORTRAN Test Program Input/Output Files

<u>PROGRAM</u>	<u>INPUT/OUTPUT FILES(FILENAME-UNIT NUMBER)</u>				
FFDE220	OFDE201-11	OFDE202-12	OFDE203-13	OFDE204-14	OFDE205-15
	OFDE206-16	OFDE207-17	OFDE208-18	OFDE209-19	OFDE20A-20
	OFDE211-21	OFDE212-22	OFDE213-23	OFDE214-24	OFDE215-25
	OFDE216-26				
FFDE250	IFDE231-11	IFDE232-12	IFDE233-13	IFDE234-14	IFDE235-15
	IFDE236-16	IFDE237-17	IFDE238-18	IFDE239-19	IFDE23A-20
	IFDE241-21	IFDE242-22	IFDE243-23	IFDE244-24	IFDE245-25
	IFDE246-26				
FFDE420	OFDE401-11	OFDE402-12	OFDE403-13	OFDE404-14	OFDE405-15
	OFDE406-16	OFDE407-17	OFDE408-18	OFDE409-19	OFDE40A-20
	OFDE411-21	OFDE412-22	OFDE413-23	OFDE414-24	OFDE415-25
	OFDE416-26				
FFDE450	IFDE431-11	IFDE432-12	IFDE433-13	IFDE434-14	IFDE435-15
	IFDE436-16	IFDE437-17	IFDE438-18	IFDE439-19	IFDE43A-20
	IFDE441-21	IFDE442-22	IFDE443-23	IFDE444-24	IFDE445-25
	IFDE446-26				

Table 7-3. Test Support Software File Types

File Name	File Type									
	Ada	CMN	CPL	DAT	DEC	FOR	JOV	SPA	TIM	USE
<u>Ada Test Support Files</u>										
ADADMP	X									
ADASPC										
ADASYS										X
ADATYP										X
MATHFUN					X					
OURDMP	X									
OURSPC								X	X	
OURSYS	X							X	X	
OURSYSR	X							X	X	
OURTYP								X	X	
OURTYPR								X	X	
<u>JOVIAL Test Support Files</u>										
DMPJOV							X			
JOVDMP			X							
JOVMATH				X						
JOVSPC			X							
JOVSYS							X			
JOVTYP										X
OURJOV			X							
SPCJOV							X			
SYSJOV						X				
TMPDMP						X				
TYPJOV										X
<u>FORTRAN Test Support Files</u>										
FORDMP						X				
FORSYS						X				
FORTYP										X
OURFOR		X								
SYSBLK						X				
SYSFOR		X								
TYPFOR										X

Table 7-4. Language-Specific Test Support Software Files

<u>File Name</u>			<u>Description</u>
<u>Ada</u>	<u>JOVIAL</u>	<u>FORTRAN</u>	
OURSYS	OURJOV	OURFOR SYSBLK	Package specification for numeric types, test control and reporting procedures, and scalar global variables.
ADASYS	JOVSYS SYSJOV	SYSFOR FORSYS	Package body implementing procedures defined in OURSYS.
OURSYSR			Same as OURSYS except range constraints are supplied for each numeric data type.
OURDMP	JOVDMP		Package specification for global variable dump procedures.
ADADMP	DMPJOV TMPDMP	FORDMP	Package body implementing procedures defined in OURDMP.
OURTYP	JOVTYP	FORTYP	Package specifications for special numeric types and global variables, and for procedures that set/use them.
OURTYPR			Same as OURTYP except range constraints are supplied for each numeric data type.
ADATYP	TYPJOV	TYPFOR	Package body implementing procedures defined in OURTYP.
OURSPC	JOVSPC		Package specification for composite data types and global variables and procedures that set/use them.
ADASPC	SPCJOV		Package body implementing procedures defined in OURSPC
MATHFUN	JOVMATH		Package specifying standard names for math functions as referenced in the WHETSTONE benchmark.

8. Ada RUN-TIME ENVIRONMENT PERFORMANCE EVALUATION

The purpose of the ACPS test evaluation capability is to provide a mechanism to assist assessment of the performance impact of Ada run-time environments. An extensive set of test cases is provided in Ada along with a fewer number of functionally equivalent JOVIAL and FORTRAN tests. Five versions of each Ada test are supplied to reveal the effect of constraint checking, of exception handlers, and of various optimization options. The majority of tests are designed to exercise individual features of the Ada language. A consistent test methodology was developed and used to ensure to the extent possible that performance measurements taken are complete, accurate, repeatable, and isolated from irrelevant factors. A common language test support interface was defined which permitted development of a capability to automatically compare the run-time performance of different compilers across the three test languages.

Performance evaluations of Ada run-time environments must be made in accordance with the requirements of specific applications. Performance issues and selection criteria for Ada run-time environments are dependent not only upon the specific application concerns but on the general type of application. For example, for applications on computers with general purpose operating systems, the performance of Ada tasking may be of little concern. For DEC VAX/VMS, commonly available Ada implementations map all Ada tasks in an executing program into the same VMS process. Due to the virtual memory nature of VMS, this is an inherently inefficient implementation in that a single disk page-fault will halt all Ada tasks from executing until the fault is satisfied. The key performance concern will not be how efficiently tasking is implemented within a single VMS process but whether the Ada compiler permits use of VMS tasking system services. On the other hand, for Ada implementations on target machines such as the MIL-STD 1750A with no underlying executive or operating system, the implementation of Ada tasking will be of paramount concern.

Functionality and space/time requirements of capabilities provided are two aspects of run-time performance. There are many compiler-dependent features in Ada which are tested in isolation by the ACPS. If these features (e.g., INLINE pragma, representation specifications) are not supported by a compiler, then for certain applications, that compiler can be considered on an a priori basis to be too inefficient to be used. The ACPS only tests features of Ada that are described in the LRM. Some compilers may provide additional pragmas and packages that could have a significant positive impact on performance for specific applications. These compiler-dependent performance enhancements must be considered in any performance evaluation of Ada run-time environments.

The ACPS consists of black box tests designed to test language feature implementation rather than compiler implementation. Just as it is practically impossible to develop black box tests to debug compilers, it is also practically impossible to develop black box benchmarks to explore all performance concerns for a given Ada compiler. Additional tests will need to be developed to investigate the performance ramifications of individual run-time system architectures.

In light of the above comments, it is still our perception that the ACPS test evaluation capability can play an important role in Ada run-time environment evaluations. The performance measures provided can give the user a good first cut estimate of the run-time efficiency of an Ada implementation. They can also furnish the user with a basis for understanding the performance behavior of application oriented benchmarks. Numerous comparison options are provided. One can automatically compare Ada implementations for the same machine architecture to provide quantifiable evidence to justify selection of a clearly superior implementation or to identify several implementations for detailed consideration. One can also get rough estimates of the resource impact of using Ada versus FORTRAN and JOVIAL for those language features for which functionally equivalent tests exist within the ACPS.

In relating performance evaluation to a specific application, a user can select various tests from the ACPS and by varying the test duration can generate a composite test sequence modeling the expected frequency of language feature usage within the application. This composite test sequence can then be used with the test comparison capability to compare the performance of different Ada implementations. The ACPS can also be used in applications planning to demonstrate which features of Ada are inefficiently implemented. This information can then be used in several ways. It can be used as justification for avoiding certain language features or processing options. It can focus attention to those areas of the compiler that must be improved and provide a mechanism for measuring improvements or degradations in run-time performance as a compiler implementation evolves. The ACPS can also be used to assist determination as to whether existing JOVIAL or FORTRAN modules should be converted to Ada by showing the probable performance impact of straightforward translation to Ada as would likely be obtained through use of a language translation tool.

APPENDIX A
ANSI TAPE FORMAT

Four tapes are written in ANSI ASCII "a" character format as volumes ACPS, ACPS02, ACPS03, ACPS04, and ACPS05. The tapes are structured as a multi-file, multi-volume configuration. The different files represent collections of ACPS test programs. Each collection of similar files is grouped together on the tape. Tables A-1 through A-5 contain a complete list of the files in the order they appear on each tape. The ordering of files in a row of each figure is from left to right.

The tapes were written in a VAX/VMS environment at a density of 1600 bpi using the copy command.

Each tape has the following format:

14 - 18 feet of blank tape
BOT (Beginning Of Tape) mark
VOL1 label = ACPS

HDR1 label
HDR2 label
TM (Tape Mark) - start of file
<<file 1 block described below>>
TM - end of file
EOF1 label
EOF2 label

TM - inter-file

HDR1 label
HDR2 label
TM - end of file
EOF1 label
EOF2 label

TM - inter-file
... other file blocks ...
TM - inter-file

HDR1 label
HDR2 label
TM
<<file n blocks described below>>
TM
EOF1 label
EOF2 label

TM - end of tape 1
TM - end of tape 2
Scratch Tape
EOT (End Of Tape) mark

The HDR1 label is a file description block providing file identifier information and creation dates.

The HDR2 label is a record description block describing the record format of the specific file.

The file blocks are at most 1024 characters in length. Each record within a file block represents a line of text. The first four characters of each record indicate the length of the record. This length includes the four-character length count. For example, a blank line would be represented as "0004" and a line containing the characters "ADA" would be represented as "0007ADA."

The files on the tapes are organized as follows:

Ada test files on volumes ACPS, ACPS02, APC503, and ACPS04
FORTRAN test files on volumes ACPS04 and ACPS05
JOVIAL test files on volume ACPS05
Test comparison tool files on volume ACPS05

Table A-1 ACPS ANSI Tape Contents Volume ACPS

AA00000.ADA	AA0P000.ADA	ADADMP.ADA	ADAPARM.ADA	ADASPC.ADA	AF03500.ADA
AF03501.ADA	AF03502.ADA	AF03503.ADA	AF03504.ADA	AF03509.ADA	AF03510.ADA
AF03511.ADA	AF03512.ADA	AF03513.ADA	AF03514.ADA	AF03517.ADA	AF03518.ADA
AF03519.ADA	AF03550.ADA	AF03600.ADA	AF03601.ADA	AF03602.ADA	AF03603.ADA
AF03604.ADA	AF03605.ADA	AF03606.ADA	AF03607.ADA	AF03608.ADA	AF03609.ADA
AF03610.ADA	AF03611.ADA	AF03612.ADA	AF03613.ADA	AF03614.ADA	AF03615.ADA
AF03616.ADA	AF03617.ADA	AF03618.ADA	AF03619.ADA	AF03620.ADA	AF03630.ADA
AF03631.ADA	AF03632.ADA	AF03633.ADA	AF03634.ADA	AF03635.ADA	AF03636.ADA
AF03637.ADA	AF03638.ADA	AF03639.ADA	AF03641.ADA	AF03642.ADA	AF03643.ADA
AF03644.ADA	AF03645.ADA	AF03646.ADA	AF03647.ADA	AF03648.ADA	AF03649.ADA
AF03650.ADA	AF03700.ADA	AF03701.ADA	AF03702.ADA	AF03703.ADA	AF03704.ADA
AF03800.ADA	AF03801.ADA	AF03802.ADA	AF03803.ADA	AF03804.ADA	AF03805.ADA
AF04120.ADA	AF04121.ADA	AF04122.ADA	AF04123.ADA	AF04124.ADA	AF04125.ADA
AF04126.ADA	AF04127.ADA	AF04129.ADA	AF0412B.ADA	AF04130.ADA	AF04131.ADA
AF04132.ADA	AF04133.ADA	AF0413D.ADA	AF0413H.ADA	AF0413I.ADA	AF0413J.ADA
AF0413K.ADA	AF04310.ADA	AF04311.ADA	AF04312.ADA	AF04510.ADA	AF04511.ADA
AF04512.ADA	AF04513.ADA	AF04514.ADA	AF04515.ADA	AF04516.ADA	AF04517.ADA
AF04518.ADA	AF04519.ADA	AF0451A.ADA	AF0451B.ADA	AF0451C.ADA	AF0451D.ADA
AF0451E.ADA	AF0451F.ADA	AF0451G.ADA	AF0451H.ADA	AF0451I.ADA	AF0451J.ADA
AF0451V.ADA	AF0451W.ADA	AF0451X.ADA	AF0451Y.ADA	AF0451Z.ADA	AF04520.ADA
AF04521.ADA	AF04522.ADA	AF04527.ADA	AF04529.ADA	AF04530.ADA	AF04531.ADA
AF04532.ADA	AF04533.ADA	AF04534.ADA	AF04535.ADA	AF04536.ADA	AF04537.ADA
AF04538.ADA	AF04539.ADA	AF0453A.ADA	AF0453B.ADA	AF0453C.ADA	AF04540.ADA
AF04541.ADA	AF04550.ADA	AF04551.ADA	AF04552.ADA	AF04553.ADA	AF04554.ADA
AF04555.ADA	AF04556.ADA	AF04557.ADA	AF04558.ADA	AF04559.ADA	AF0455N.ADA
AF04550.ADA	AF04560.ADA	AF04562.ADA	AF04563.ADA	AF04600.ADA	AF04601.ADA
AF04602.ADA	AF0460E.ADA	AF0460F.ADA	AF0460G.ADA	AF0460H.ADA	AF0460I.ADA
AF0460J.ADA	AF0460M.ADA	AF05200.ADA	AF05201.ADA	AF05202.ADA	AF05203.ADA
AF05204.ADA	AF05205.ADA	AF05206.ADA	AF05207.ADA	AF05208.ADA	AF05209.ADA
AF0520A.ADA	AF0520B.ADA	AF0520M.ADA	AF05200.ADA	AF0520R.ADA	AF0520S.ADA
AF05210.ADA	AF05211.ADA	AF05212.ADA	AF05213.ADA	AF05214.ADA	AF05215.ADA
AF0521H.ADA	AF0521I.ADA	AF0521J.ADA	AF0521K.ADA	AF0521L.ADA	AF0521N.ADA
AF05300.ADA	AF05301.ADA	AF05302.ADA	AF05303.ADA	AF05304.ADA	AF05305.ADA
AF05306.ADA	AF05307.ADA	AF05308.ADA	AF05400.ADA	AF05401.ADA	AF05402.ADA
AF05403.ADA	AF05404.ADA	AF05405.ADA	AF05406.ADA	AF05407.ADA	AF05408.ADA
AF05501.ADA	AF05502.ADA	AF05503.ADA	AF05504.ADA	AF05505.ADA	AF05506.ADA
AF05507.ADA	AF05508.ADA	AF05509.ADA	AF0550A.ADA	AF0550B.ADA	AF0550C.ADA
AF0550D.ADA	AF0550E.ADA	AF0550F.ADA	AF06001.ADA	AF06009.ADA	AF06010.ADA
AF06011.ADA	AF06013.ADA	AF06014.ADA	AF06015.ADA	AF06016.ADA	AF06017.ADA
AF06018.ADA	AF06019.ADA	AF06022.ADA	AF06023.ADA	AF06024.ADA	AF06025.ADA
AF06026.ADA	AF06027.ADA	AF06028.ADA	AF06029.ADA	AF06030.ADA	AF06031.ADA
AF06032.ADA	AF06033.ADA	AF06043.ADA	AF06044.ADA	AF06045.ADA	AF06046.ADA
AF06047.ADA	AF06048.ADA	AF06049.ADA	AF06050.ADA	AF06051.ADA	AF06052.ADA
AF06053.ADA	AF06060.ADA	AF06061.ADA	AF06062.ADA	AF06063.ADA	AF06064.ADA
AF06065.ADA	AF06066.ADA	AF06067.ADA	AF06068.ADA	AF06069.ADA	AF06070.ADA
AF06071.ADA	AF06072.ADA	AF06073.ADA	AF06074.ADA	AF06075.ADA	AF06076.ADA
AF06077.ADA	AF06078.ADA	AF06079.ADA	AF06100.ADA	AF06101.ADA	AF06108.ADA
AF06109.ADA	AF06110.ADA	AF06111.ADA	AF06112.ADA	AF06113.ADA	AF06114.ADA
AF06115.ADA	AF06116.ADA	AF06117.ADA	AF06118.ADA	AF06119.ADA	AF06122.ADA
AF06132.ADA	AF06133.ADA	AF06134.ADA	AF06135.ADA	AF06136.ADA	AF06137.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

AF06817.ADA	AF06818.ADA	AF06819.ADA	AF06820.ADA	AF06821.ADA	AF06822.ADA
AF06823.ADA	AF06824.ADA	AF06825.ADA	AF09500.ADA	AF09501.ADA	AF09502.ADA
AF09503.ADA	AF09504.ADA	AF09505.ADA	AF09506.ADA	AF09507.ADA	AF09508.ADA
AF09509.ADA	AF09600.ADA	AF09601.ADA	AF09602.ADA	AF09603.ADA	AF09604.ADA
AF09605.ADA	AF09606.ADA	AF09607.ADA	AF09710.ADA	AF09711.ADA	AF09712.ADA
AF09713.ADA	AF09720.ADA	AF09721.ADA	AF09731.ADA	AF09900.ADA	AF09901.ADA
AF09902.ADA	AF09903.ADA	AF09B01.ADA	AF0C100.ADA	AF0C101.ADA	AF0C102.ADA
AF0C103.ADA	AF0C300.ADA	AF0C301.ADA	AF0C302.ADA	AF0C303.ADA	AF0C304.ADA
AF0C305.ADA	AF0C306.ADA	AF0C307.ADA	AF0D720.ADA	AF0D721.ADA	AF0D727.ADA
AF0D728.ADA	AF0D729.ADA	AF0D72D.ADA	AF0P000.ADA	AFD3551.ADA	AFD412A.ADA
AFD412C.ADA	AFD412D.ADA	AFD4135.ADA	AFD4136.ADA	AFD4137.ADA	AFD4138.ADA
AFD4139.ADA	AFD413A.ADA	AFD413B.ADA	AFD413C.ADA	AFD413E.ADA	AFD413G.ADA
AFD451K.ADA	AFD451L.ADA	AFD451M.ADA	AFD451N.ADA	AFD451O.ADA	AFD451P.ADA
AFD451Q.ADA	AFD451R.ADA	AFD451S.ADA	AFD4523.ADA	AFD4524.ADA	AFD4525.ADA
AFD4526.ADA	AFD4528.ADA	AFD452A.ADA	AFD455D.ADA	AFD455E.ADA	AFD455F.ADA
AFD455G.ADA	AFD455I.ADA	AFD455J.ADA	AFD455K.ADA	AFD455L.ADA	AFD4556.ADA
AFD4567.ADA	AFD4568.ADA	AFD4569.ADA	AFD4603.ADA	AFD4604.ADA	AFD4605.ADA
AFD4606.ADA	AFD4607.ADA	AFD4608.ADA	AFD4609.ADA	AFD460A.ADA	AFD460B.ADA
AFD460K.ADA	AFD460L.ADA	AFD460N.ADA	AFD460O.ADA	AFD4800.ADA	AFD4801.ADA
AFD4802.ADA	AFD4803.ADA	AFD4804.ADA	AFD520C.ADA	AFD520D.ADA	AFD520E.ADA
AFD520F.ADA	AFD520G.ADA	AFD520H.ADA	AFD520I.ADA	AFD520J.ADA	AFD520N.ADA
AFD520P.ADA	AFD520Q.ADA	AFD5218.ADA	AFD5219.ADA	AFD521A.ADA	AFD521B.ADA
AFD521C.ADA	AFD521D.ADA	AFD521E.ADA	AFD521F.ADA	AFD521K.ADA	AFD521M.ADA
AFD521O.ADA	AFD521P.ADA	AFD6201.ADA	AFD6209.ADA	AFD6210.ADA	AFD6211.ADA
AFD6213.ADA	AFD6214.ADA	AFD6215.ADA	AFD6216.ADA	AFD6217.ADA	AFD6218.ADA
AFD6219.ADA	AFD6222.ADA	AFD6223.ADA	AFD6224.ADA	AFD6225.ADA	AFD6226.ADA
AFD6227.ADA	AFD6228.ADA	AFD6229.ADA	AFD6230.ADA	AFD6231.ADA	AFD6232.ADA
AFD6233.ADA	AFD6243.ADA	AFD6244.ADA	AFD6245.ADA	AFD6246.ADA	AFD6247.ADA
AFD6248.ADA	AFD6249.ADA	AFD6250.ADA	AFD6251.ADA	AFD6252.ADA	AFD6253.ADA
AFD6260.ADA	AFD6261.ADA	AFD6262.ADA	AFD6263.ADA	AFD6264.ADA	AFD6265.ADA
AFD6266.ADA	AFD6267.ADA	AFD6268.ADA	AFD6269.ADA	AFD6270.ADA	AFD6271.ADA
AFD6272.ADA	AFD6273.ADA	AFD6274.ADA	AFD6275.ADA	AFD6276.ADA	AFD6277.ADA
AFD6278.ADA	AFD6279.ADA	AFD6300.ADA	AFD6301.ADA	AFD6308.ADA	AFD6309.ADA
AFD6310.ADA	AFD6311.ADA	AFD6312.ADA	AFD6313.ADA	AFD6314.ADA	AFD6315.ADA
AFD6316.ADA	AFD6317.ADA	AFD6318.ADA	AFD6319.ADA	AFD6322.ADA	AFD6323.ADA
AFD6333.ADA	AFD6334.ADA	AFD6335.ADA	AFD6336.ADA	AFD6337.ADA	AFD6338.ADA
AFD6339.ADA	AFD6340.ADA	AFD6342.ADA	AFD6343.ADA	AFD6344.ADA	AFD6345.ADA
AFD6346.ADA	AFD6347.ADA	AFD6348.ADA	AFD6349.ADA	AFD6350.ADA	AFD6352.ADA
AFD6353.ADA	AFD6354.ADA	AFD6355.ADA	AFD6356.ADA	AFD6357.ADA	AFD6358.ADA
AFD6359.ADA	AFD6360.ADA	AFD6362.ADA	AFD6363.ADA	AFD6364.ADA	AFD6365.ADA
AFD6366.ADA	AFD6367.ADA	AFD6368.ADA	AFD6369.ADA	AFD6370.ADA	AFD6372.ADA
AFD6373.ADA	AFD6374.ADA	AFD6375.ADA	AFD6376.ADA	AFD6377.ADA	AFD6378.ADA
AFD6379.ADA	AFD6380.ADA	AFD6382.ADA	AFD6383.ADA	AFD6384.ADA	AFD6385.ADA
AFD6386.ADA	AFD6387.ADA	AFD6388.ADA	AFD6389.ADA	AFD6390.ADA	AFD680G.ADA
AFD680H.ADA	AFD680I.ADA	AFD680J.ADA	AFD9000.ADA	AFD9200.ADA	AFD9C00.ADA
AFD9C01.ADA	AFD9C02.ADA	AFDC104.ADA	AFDC105.ADA	AFDC106.ADA	AFDC107.ADA
AFDC310.ADA	AFDC311.ADA	AFDC313.ADA	AFDC314.ADA	AFDD600.ADA	AFDD601.ADA
AFDD602.ADA	AFDD603.ADA	AFDD604.ADA	AFDD605.ADA	AFDD606.ADA	AFDD607.ADA
AFDD609.ADA	AFDD610.ADA	AFDD611.ADA	AFDD722.ADA	AFDD723.ADA	AFDD724.ADA
AFDD725.ADA	AFDD72A.ADA	AFDD72B.ADA	AFDD72C.ADA	AFDD72E.ADA	AFDD72F.ADA
AFDD72G.ADA	AFDD72H.ADA	AFDD72I.ADA	AFDDA01.ADA	AFDDA02.ADA	AFDE000.ADA
AFDE201.ADA	AFDE202.ADA	AFDE203.ADA	AFDE204.ADA	AFDE205.ADA	AFDE206.ADA
AFDE207.ADA	AFDE208.ADA	AFDE209.ADA	AFDE20A.ADA	AFDE211.ADA	AFDE212.ADA
AFDE213.ADA	AFDE214.ADA	AFDE215.ADA	AFDE216.ADA	AFDE220.ADA	AFDE231.ADA
AFDE232.ADA	AFDE233.ADA	AFDE234.ADA	AFDE235.ADA	AFDE236.ADA	AFDE237.ADA
AFDE238.ADA	AFDE239.ADA	AFDE23A.ADA	AFDE241.ADA	AFDE242.ADA	AFDE243.ADA
AFDE244.ADA	AFDE245.ADA	AFDE246.ADA	AFDE250.ADA	AFDE401.ADA	AFDE402.ADA
AFDE403.ADA	AFDE404.ADA	AFDE405.ADA	AFDE406.ADA	AFDE407.ADA	AFDE408.ADA
AFDE409.ADA	AFDE40A.ADA	AFDE411.ADA	AFDE412.ADA	AFDE413.ADA	AFDE414.ADA
AFDE415.ADA	AFDE416.ADA	AFDE420.ADA	AFDE431.ADA	AFDE432.ADA	AFDE433.ADA
AFDE434.ADA	AFDE435.ADA	AFDE436.ADA	AFDE437.ADA	AFDE438.ADA	AFDE439.ADA
AFDE43A.ADA	AFDE441.ADA	AFDE442.ADA	AFDE443.ADA	AFDE444.ADA	AFDE445.ADA
AFDE446.ADA	AFDE450.ADA	AFDF000.ADA	AFDF001.ADA	AFDF002.ADA	AFDF003.ADA
AFDF004.ADA	AFDF005.ADA	AFIRST.ADA	AFM9A00.ADA	AFM9A01.ADA	AFM9A02.ADA
AFM9A03.ADA	AFMB000.ADA	AFMB001.ADA	AFMB002.ADA	AFMB003.ADA	AFMB004.ADA
AFMB005.ADA	AFMB006.ADA	AFMB007.ADA	AFMB008.ADA	AFMB009.ADA	AFMB010.ADA
AFMB011.ADA	AFMB012.ADA	AFMB013.ADA	AFMB014.ADA	AFMB015.ADA	AFMB016.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

AFMB017.ADA	AFMB018.ADA	AFMB019.ADA	AFMB020.ADA	AFMB021.ADA	AFMB022.ADA
AFMB023.ADA	AFMB024.ADA	AFMB025.ADA	AFMB026.ADA	AFMB027.ADA	AFMB028.ADA
AFMB029.ADA	AFMB030.ADA	AFMB031.ADA	AFMB032.ADA	AFMB033.ADA	AFMB034.ADA
AFMB035.ADA	AFMB036.ADA	AFMB037.ADA	AFMB038.ADA	AFMB039.ADA	AFMB040.ADA
AFMB041.ADA	AFMB042.ADA	AFMB043.ADA	AFMB044.ADA	AFN9300.ADA	AFN9301.ADA
AFN9302.ADA	AFN9303.ADA	AFN9510.ADA	AFN9511.ADA	AFN9610.ADA	AFN9611.ADA
AFN9612.ADA	AFN9613.ADA	AGD0000C.ADA	AGD0000.ADA	AGD0000.ADA	AGD0001.ADA
AGD0002.ADA	AGD0003.ADA	AGD0004.ADA	AGD0005.ADA	AGD0006.ADA	AGD0007.ADA
AGD0008.ADA	AGD0009.ADA	AGD000D.ADA	AGD000E.ADA	AGD0019.ADA	AGD0029.ADA
AGD0039.ADA	AGD0049.ADA	AGN000F.ADA	AGN000G.ADA	AGN000H.ADA	AGN000I.ADA
AGN000J.ADA	AGN000K.ADA	AL09000.ADA	AL09100.ADA	AL09101.ADA	AL09111.ADA
AL09121.ADA	AL09131.ADA	AL0P000.ADA	ALD9200.ADA	ALD9202.ADA	ALD9203.ADA
ALD9212.ADA	ALD9213.ADA	ALD9222.ADA	ALD9223.ADA	ALD9232.ADA	ALD9233.ADA
ALDE000.ADA	ALDE2B1.ADA	ALDE2B2.ADA	ALDE2C1.ADA	ALDE2C2.ADA	ALDE2C5.ADA
ALDE2C6.ADA	ALDE4B1.ADA	ALDE4B2.ADA	ALDE4C1.ADA	ALDE4C2.ADA	ALDE4C5.ADA
ALDE4C6.ADA	A000001.ADA	A000002.ADA	A000102.ADA	A000103.ADA	A000104.ADA
A000107.ADA	A000108.ADA	A000109.ADA	A000112.ADA	A000113.ADA	A000114.ADA
A000202.ADA	A000203.ADA	A000204.ADA	A000207.ADA	A000208.ADA	A000209.ADA
A000300.ADA	A000305.ADA	A000310.ADA	A000311.ADA	A000312.ADA	A000313.ADA
A000314.ADA	A000315.ADA	A000316.ADA	A000317.ADA	A000318.ADA	A000319.ADA
A000320.ADA	A000321.ADA	A000322.ADA	A000323.ADA	A000324.ADA	A000325.ADA
A000326.ADA	A000327.ADA	A000328.ADA	A000329.ADA	A000330.ADA	A000331.ADA
A000332.ADA	A000333.ADA	A000334.ADA	A000335.ADA	A000336.ADA	A000337.ADA
A000338.ADA	A000339.ADA	A000340.ADA	A000341.ADA	A000342.ADA	A000343.ADA
A000347.ADA	A000348.ADA	A000349.ADA	A000350.ADA	A000351.ADA	A000352.ADA
A000353.ADA	A000354.ADA	A000355.ADA	A000356.ADA	A000357.ADA	A000358.ADA
A000359.ADA	A000360.ADA	A000361.ADA	A000362.ADA	A000363.ADA	A000364.ADA
A000365.ADA	A000366.ADA	A000367.ADA	A000368.ADA	A000369.ADA	A00036A.ADA
A00036B.ADA	A00036F.ADA	A00036G.ADA	A00036H.ADA	A00036I.ADA	A00036J.ADA
A00036K.ADA	A00036L.ADA	A00036M.ADA	A00036N.ADA	A00036U.ADA	A00036V.ADA
A00036W.ADA	A000374.ADA	A000375.ADA	A000376.ADA	A000377.ADA	A000378.ADA
A000390.ADA	A000391.ADA	A000402.ADA	A000403.ADA	A000404.ADA	A000500.ADA
A000502.ADA	A000503.ADA	A000504.ADA	A000507.ADA	A000508.ADA	A000509.ADA
A000512.ADA	A000513.ADA	A000514.ADA	A000517.ADA	A000518.ADA	A000519.ADA
A000602.ADA	A000603.ADA	A000604.ADA	A000605.ADA	A000606.ADA	A000607.ADA
A000702.ADA	A000703.ADA	A000704.ADA	A000705.ADA	A000706.ADA	A000707.ADA
A000708.ADA	A000709.ADA	A000710.ADA	A000711.ADA	A000713.ADA	A000714.ADA
A000715.ADA	A000716.ADA	A000717.ADA	A000718.ADA	A000719.ADA	A000720.ADA
A000721.ADA	A00P000.ADA	A0D0344.ADA	A0D0345.ADA	A0D0346.ADA	A0D036C.ADA
A0D036D.ADA	A0D036E.ADA	A0D0360.ADA	A0D036P.ADA	A0D036Q.ADA	A0D036R.ADA
A0D036S.ADA	A0D036T.ADA	A0D036X.ADA	A0D036Y.ADA	A0D036Z.ADA	A0D0370.ADA
A0D0371.ADA	A0D0372.ADA	A0D0373.ADA	A0D0380.ADA	A0D0381.ADA	A0D0382.ADA
A0D0383.ADA	A0D0523.ADA	A0D0524.ADA	A0D0525.ADA	A0D0526.ADA	A0D0527.ADA
A0D0528.ADA	A0D0529.ADA	CA00000.ADA	CA0P000.ADA	CF03500.ADA	CF03501.ADA
CF03502.ADA	CF03503.ADA	CF03504.ADA	CF03509.ADA	CF03510.ADA	CF03511.ADA
CF03512.ADA	CF03513.ADA	CF03514.ADA	CF03517.ADA	CF03518.ADA	CF03519.ADA
CF03550.ADA	CF03600.ADA	CF03601.ADA	CF03602.ADA	CF03603.ADA	CF03604.ADA
CF03605.ADA	CF03606.ADA	CF03607.ADA	CF03608.ADA	CF03609.ADA	CF03610.ADA
CF03611.ADA	CF03612.ADA	CF03613.ADA	CF03614.ADA	CF03615.ADA	CF03616.ADA
CF03617.ADA	CF03618.ADA	CF03619.ADA	CF03620.ADA	CF03630.ADA	CF03631.ADA
CF03632.ADA	CF03633.ADA	CF03634.ADA	CF03635.ADA	CF03636.ADA	CF03637.ADA
CF03638.ADA	CF03639.ADA	CF03641.ADA	CF03642.ADA	CF03643.ADA	CF03644.ADA
CF03645.ADA	CF03646.ADA	CF03647.ADA	CF03648.ADA	CF03649.ADA	CF03650.ADA
CF03700.ADA	CF03701.ADA	CF03702.ADA	CF03703.ADA	CF03704.ADA	CF03800.ADA
CF03801.ADA	CF03802.ADA	CF03803.ADA	CF03804.ADA	CF03805.ADA	CF04120.ADA
CF04121.ADA	CF04122.ADA	CF04123.ADA	CF04124.ADA	CF04125.ADA	CF04126.ADA
CF04127.ADA	CF04129.ADA	CF0412B.ADA	CF04130.ADA	CF04131.ADA	CF04132.ADA
CF04133.ADA	CF0413D.ADA	CF0413H.ADA	CF0413I.ADA	CF0413J.ADA	CF0413K.ADA
CF04310.ADA	CF04311.ADA	CF04312.ADA	CF04510.ADA	CF04511.ADA	CF04512.ADA
CF04513.ADA	CF04514.ADA	CF04515.ADA	CF04516.ADA	CF04517.ADA	CF04518.ADA
CF04519.ADA	CF0451A.ADA	CF0451B.ADA	CF0451C.ADA	CF0451D.ADA	CF0451E.ADA
CF0451F.ADA	CF0451G.ADA	CF0451H.ADA	CF0451I.ADA	CF0451J.ADA	CF0451V.ADA
CF0451W.ADA	CF0451X.ADA	CF0451Y.ADA	CF0451Z.ADA	CF04520.ADA	CF04521.ADA
CF04522.ADA	CF04527.ADA	CF04529.ADA	CF04530.ADA	CF04531.ADA	CF04532.ADA
CF04533.ADA	CF04534.ADA	CF04535.ADA	CF04536.ADA	CF04537.ADA	CF04538.ADA
CF04539.ADA	CF0453A.ADA	CF0453B.ADA	CF0453C.ADA	CF04540.ADA	CF04541.ADA
CF04550.ADA	CF04551.ADA	CF04552.ADA	CF04553.ADA	CF04554.ADA	CF04555.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

[illegible]

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

CFD6222.ADA	CFD6223.ADA	CFD6224.ADA	CFD6225.ADA	CFD6226.ADA	CFD6227.ADA
CFD6228.ADA	CFD6229.ADA	CFD6230.ADA	CFD6231.ADA	CFD6232.ADA	CFD6233.ADA
CFD6243.ADA	CFD6244.ADA	CFD6245.ADA	CFD6246.ADA	CFD6247.ADA	CFD6248.ADA
CFD6249.ADA	CFD6250.ADA	CFD6251.ADA	CFD6252.ADA	CFD6253.ADA	CFD6260.ADA
CFD6261.ADA	CFD6262.ADA	CFD6263.ADA	CFD6264.ADA	CFD6265.ADA	CFD6266.ADA
CFD6267.ADA	CFD6268.ADA	CFD6269.ADA	CFD6270.ADA	CFD6271.ADA	CFD6272.ADA
CFD6273.ADA	CFD6274.ADA	CFD6275.ADA	CFD6276.ADA	CFD6277.ADA	CFD6278.ADA
CFD6279.ADA	CFD6300.ADA	CFD6301.ADA	CFD6308.ADA	CFD6309.ADA	CFD6310.ADA
CFD6311.ADA	CFD6312.ADA	CFD6313.ADA	CFD6314.ADA	CFD6315.ADA	CFD6316.ADA
CFD6317.ADA	CFD6318.ADA	CFD6319.ADA	CFD6322.ADA	CFD6332.ADA	CFD6333.ADA
CFD6334.ADA	CFD6335.ADA	CFD6336.ADA	CFD6337.ADA	CFD6338.ADA	CFD6339.ADA
CFD6340.ADA	CFD6342.ADA	CFD6343.ADA	CFD6344.ADA	CFD6345.ADA	CFD6346.ADA
CFD6347.ADA	CFD6348.ADA	CFD6349.ADA	CFD6350.ADA	CFD6352.ADA	CFD6353.ADA
CFD6354.ADA	CFD6355.ADA	CFD6356.ADA	CFD6357.ADA	CFD6358.ADA	CFD6359.ADA
CFD6360.ADA	CFD6362.ADA	CFD6363.ADA	CFD6364.ADA	CFD6365.ADA	CFD6366.ADA
CFD6367.ADA	CFD6368.ADA	CFD6369.ADA	CFD6370.ADA	CFD6372.ADA	CFD6373.ADA
CFD6374.ADA	CFD6375.ADA	CFD6376.ADA	CFD6377.ADA	CFD6378.ADA	CFD6379.ADA
CFD6380.ADA	CFD6382.ADA	CFD6383.ADA	CFD6384.ADA	CFD6385.ADA	CFD6386.ADA
CFD6387.ADA	CFD6388.ADA	CFD6389.ADA	CFD6390.ADA	CFD680G.ADA	CFD680H.ADA
CFD680I.ADA	CFD680J.ADA	CFD9000.ADA	CFD9200.ADA	CFD9C00.ADA	CFD9C01.ADA
CFD9C02.ADA	CFDC104.ADA	CFDC105.ADA	CFDC106.ADA	CFDC107.ADA	CFDC310.ADA
CFDC311.ADA	CFDC313.ADA	CFDC314.ADA	CFDD600.ADA	CFDD601.ADA	CFDD602.ADA
CFDD603.ADA	CFDD604.ADA	CFDD605.ADA	CFDD606.ADA	CFDD607.ADA	CFDD609.ADA
CFDD610.ADA	CFDD611.ADA	CFDDA01.ADA	CFDDA02.ADA	CFDE000.ADA	CFDE201.ADA
CFDE202.ADA	CFDE203.ADA	CFDE204.ADA	CFDE205.ADA	CFDE206.ADA	CFDE207.ADA
CFDE208.ADA	CFDE209.ADA	CFDE20A.ADA	CFDE211.ADA	CFDE212.ADA	CFDE213.ADA
CFDE214.ADA	CFDE215.ADA	CFDE216.ADA	CFDE220.ADA	CFDE231.ADA	CFDE232.ADA
CFDE233.ADA	CFDE234.ADA	CFDE235.ADA	CFDE236.ADA	CFDE237.ADA	CFDE238.ADA
CFDE239.ADA	CFDE23A.ADA	CFDE241.ADA	CFDE242.ADA	CFDE243.ADA	CFDE244.ADA
CFDE245.ADA	CFDE246.ADA	CFDE250.ADA	CFDE401.ADA	CFDE402.ADA	CFDE403.ADA
CFDE404.ADA	CFDE405.ADA	CFDE406.ADA	CFDE407.ADA	CFDE408.ADA	CFDE409.ADA
CFDE40A.ADA	CFDE411.ADA	CFDE412.ADA	CFDE413.ADA	CFDE414.ADA	CFDE415.ADA
CFDE416.ADA	CFDE420.ADA	CFDE431.ADA	CFDE432.ADA	CFDE433.ADA	CFDE434.ADA
CFDE435.ADA	CFDE436.ADA	CFDE437.ADA	CFDE438.ADA	CFDE439.ADA	CFDE43A.ADA
CFDE441.ADA	CFDE442.ADA	CFDE443.ADA	CFDE444.ADA	CFDE445.ADA	CFDE446.ADA
CFDE450.ADA	CFDF000.ADA	CFDF001.ADA	CFDF002.ADA	CFDF003.ADA	CFDF004.ADA
CFDF005.ADA	CFM9A00.ADA	CFM9A01.ADA	CFM9A02.ADA	CFM9A03.ADA	CFM9A04.ADA
CFMB001.ADA	CFMB002.ADA	CFMB003.ADA	CFMB004.ADA	CFMB005.ADA	CFMB010.ADA
CFMB011.ADA	CFMB012.ADA	CFMB013.ADA	CFMB014.ADA	CFMB015.ADA	CFMB016.ADA
CFMB017.ADA	CFMB018.ADA	CFMB019.ADA	CFMB020.ADA	CFMB029.ADA	CFMB030.ADA
CFMB031.ADA	CFMB032.ADA	CFMB033.ADA	CFMB034.ADA	CFMB035.ADA	CFMB036.ADA
CFMB041.ADA	CFMB042.ADA	CFMB043.ADA	CFMB044.ADA	CFN9300.ADA	CFN9301.ADA
CFN9302.ADA	CFN9303.ADA	CFN9510.ADA	CFN9511.ADA	CFN9610.ADA	CFN9611.ADA
CFN9612.ADA	CFN9613.ADA	CG0000C.ADA	CG0P000.ADA	CGD0000.ADA	CGD0001.ADA
CGD0002.ADA	CGD0003.ADA	CGD0004.ADA	CGD0005.ADA	CGD0006.ADA	CGD0007.ADA
CGD0008.ADA	CGD0009.ADA	CGD000D.ADA	CGD000E.ADA	CGD0019.ADA	CGD0029.ADA
CGD0039.ADA	CGD0049.ADA	CGN000F.ADA	CGN000G.ADA	CGN000H.ADA	CGN000I.ADA
CGN000J.ADA	CGN000K.ADA	CL09000.ADA	CL09100.ADA	CL09101.ADA	CL09111.ADA
CL09121.ADA	CL09131.ADA	CL0P000.ADA	CLD9200.ADA	CLD9202.ADA	CLD9203.ADA
CLD9212.ADA	CLD9213.ADA	CLD9222.ADA	CLD9223.ADA	CLD9232.ADA	CLD9233.ADA
CLDE000.ADA	CLDE2B1.ADA	CLDE2B2.ADA	CLDE2C1.ADA	CLDE2C2.ADA	CLDE2C5.ADA
CLDE2C6.ADA	CLDE4B1.ADA	CLDE4B2.ADA	CLDE4C1.ADA	CLDE4C2.ADA	CLDE4C5.ADA
CLDE4C6.ADA	C000001.ADA	C000002.ADA	C000102.ADA	C000103.ADA	C000104.ADA
C000107.ADA	C000108.ADA	C000109.ADA	C000112.ADA	C000113.ADA	C000114.ADA
C000202.ADA	C000203.ADA	C000204.ADA	C000207.ADA	C000208.ADA	C000209.ADA
C000300.ADA	C000305.ADA	C000310.ADA	C000311.ADA	C000312.ADA	C000313.ADA
C000314.ADA	C000315.ADA	C000316.ADA	C000317.ADA	C000318.ADA	C000319.ADA
C000320.ADA	C000321.ADA	C000322.ADA	C000323.ADA	C000324.ADA	C000325.ADA
C000326.ADA	C000327.ADA	C000328.ADA	C000329.ADA	C000330.ADA	C000331.ADA
C000332.ADA	C000333.ADA	C000334.ADA	C000335.ADA	C000336.ADA	C000337.ADA
C000338.ADA	C000339.ADA	C000340.ADA	C000341.ADA	C000342.ADA	C000343.ADA
C000347.ADA	C000348.ADA	C000349.ADA	C000350.ADA	C000351.ADA	C000352.ADA
C000353.ADA	C000354.ADA	C000355.ADA	C000356.ADA	C000357.ADA	C000358.ADA
C000359.ADA	C000360.ADA	C000361.ADA	C000362.ADA	C000363.ADA	C000364.ADA
C000365.ADA	C000366.ADA	C000367.ADA	C000368.ADA	C000369.ADA	C00036A.ADA
C00036B.ADA	C00036F.ADA	C00036G.ADA	C00036H.ADA	C00036I.ADA	C00036J.ADA
C00036K.ADA	C00036L.ADA	C00036M.ADA	C00036N.ADA	C00036U.ADA	C00036V.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

C00036H.ADA	C000374.ADA	C000375.ADA	C000376.ADA	C000377.ADA	C000378.ADA
C000390.ADA	C000391.ADA	C000402.ADA	C000403.ADA	C000404.ADA	C000500.ADA
C000502.ADA	C000503.ADA	C000504.ADA	C000507.ADA	C000508.ADA	C000509.ADA
C000512.ADA	C000513.ADA	C000514.ADA	C000517.ADA	C000518.ADA	C000519.ADA
C000602.ADA	C000603.ADA	C000604.ADA	C000605.ADA	C000606.ADA	C000607.ADA
C000702.ADA	C000703.ADA	C000704.ADA	C000705.ADA	C000706.ADA	C000707.ADA
C000708.ADA	C000709.ADA	C000710.ADA	C000711.ADA	C000713.ADA	C000714.ADA
C000715.ADA	C000716.ADA	C000717.ADA	C000718.ADA	C000719.ADA	C000720.ADA
C000721.ADA	C00P000.ADA	C0D0344.ADA	C0D0345.ADA	C0D0346.ADA	C0D036C.ADA
C0D036D.ADA	C0D036E.ADA	C0D0360.ADA	C0D036P.ADA	C0D036Q.ADA	C0D036R.ADA
C0D036S.ADA	C0D036T.ADA	C0D036X.ADA	C0D036Y.ADA	C0D036Z.ADA	C0D0370.ADA
C0D0371.ADA	C0D0372.ADA	C0D0373.ADA	C0D0380.ADA	C0D0381.ADA	C0D0382.ADA
C0D0383.ADA	C0D0523.ADA	C0D0524.ADA	C0D0525.ADA	C0D0526.ADA	C0D0527.ADA
C0D0528.ADA	C0D0529.ADA	EA00000.ADA	EA0P000.ADA	EF03500.ADA	EF03501.ADA
EF03502.ADA	EF03503.ADA	EF03504.ADA	EF03509.ADA	EF03510.ADA	EF03511.ADA
EF03512.ADA	EF03513.ADA	EF03514.ADA	EF03517.ADA	EF03518.ADA	EF03519.ADA
EF03550.ADA	EF03600.ADA	EF03601.ADA	EF03602.ADA	EF03603.ADA	EF03604.ADA
EF03605.ADA	EF03606.ADA	EF03607.ADA	EF03608.ADA	EF03609.ADA	EF03610.ADA
EF03611.ADA	EF03612.ADA	EF03613.ADA	EF03614.ADA	EF03615.ADA	EF03616.ADA
EF03617.ADA	EF03618.ADA	EF03619.ADA	EF03620.ADA	EF03630.ADA	EF03631.ADA
EF03632.ADA	EF03633.ADA	EF03634.ADA	EF03635.ADA	EF03636.ADA	EF03637.ADA
EF03638.ADA	EF03639.ADA	EF03641.ADA	EF03642.ADA	EF03643.ADA	EF03644.ADA
EF03645.ADA	EF03646.ADA	EF03647.ADA	EF03648.ADA	EF03649.ADA	EF03650.ADA
EF03700.ADA	EF03701.ADA	EF03702.ADA	EF03703.ADA	EF03704.ADA	EF03800.ADA
EF03801.ADA	EF03802.ADA	EF03803.ADA	EF03804.ADA	EF03805.ADA	EF04120.ADA
EF04121.ADA	EF04122.ADA	EF04123.ADA	EF04124.ADA	EF04125.ADA	EF04126.ADA
EF04127.ADA	EF04129.ADA	EF0412B.ADA	EF04130.ADA	EF04131.ADA	EF04132.ADA
EF04133.ADA	EF0413D.ADA	EF0413H.ADA	EF0413I.ADA	EF0413J.ADA	EF0413K.ADA
EF04310.ADA	EF04311.ADA	EF04312.ADA	EF04510.ADA	EF04511.ADA	EF04512.ADA
EF04513.ADA	EF04514.ADA	EF04515.ADA	EF04516.ADA	EF04517.ADA	EF04518.ADA
EF04519.ADA	EF0451A.ADA	EF0451B.ADA	EF0451C.ADA	EF0451D.ADA	EF0451E.ADA
EF0451F.ADA	EF0451G.ADA	EF0451H.ADA	EF0451I.ADA	EF0451J.ADA	EF0451V.ADA
EF0451W.ADA	EF0451X.ADA	EF0451Y.ADA	EF0451Z.ADA	EF04520.ADA	EF04521.ADA
EF04522.ADA	EF04527.ADA	EF04529.ADA	EF04530.ADA	EF04531.ADA	EF04532.ADA
EF04533.ADA	EF04534.ADA	EF04535.ADA	EF04536.ADA	EF04537.ADA	EF04538.ADA
EF04539.ADA	EF0453A.ADA	EF0453B.ADA	EF0453C.ADA	EF04540.ADA	EF04541.ADA
EF04550.ADA	EF04551.ADA	EF04552.ADA	EF04553.ADA	EF04554.ADA	EF04555.ADA
EF04556.ADA	EF04557.ADA	EF04558.ADA	EF04559.ADA	EF0455N.ADA	EF04550.ADA
EF04560.ADA	EF04562.ADA	EF04563.ADA	EF04600.ADA	EF04601.ADA	EF04602.ADA
EF0460E.ADA	EF0460F.ADA	EF0460G.ADA	EF0460H.ADA	EF0460I.ADA	EF0460J.ADA
EF0460M.ADA	EF05200.ADA	EF05201.ADA	EF05202.ADA	EF05203.ADA	EF05204.ADA
EF05205.ADA	EF05206.ADA	EF05207.ADA	EF05208.ADA	EF05209.ADA	EF0520A.ADA
EF0520B.ADA	EF0520M.ADA	EF05200.ADA	EF0520R.ADA	EF0520S.ADA	EF05210.ADA
EF05211.ADA	EF05212.ADA	EF05213.ADA	EF05214.ADA	EF05215.ADA	EF0521H.ADA
EF0521I.ADA	EF0521J.ADA	EF0521K.ADA	EF0521L.ADA	EF0521N.ADA	EF05300.ADA
EF05301.ADA	EF05302.ADA	EF05303.ADA	EF05304.ADA	EF05305.ADA	EF05306.ADA
EF05307.ADA	EF05308.ADA	EF05400.ADA	EF05401.ADA	EF05402.ADA	EF05403.ADA
EF05404.ADA	EF05405.ADA	EF05406.ADA	EF05407.ADA	EF05408.ADA	EF05501.ADA
EF05502.ADA	EF05503.ADA	EF05504.ADA	EF05505.ADA	EF05506.ADA	EF05507.ADA
EF05508.ADA	EF05509.ADA	EF0550A.ADA	EF0550B.ADA	EF0550C.ADA	EF0550D.A

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

EF06166.ADA	EF06167.ADA	EF06168.ADA	EF06169.ADA	EF06170.ADA	EF06172.ADA
EF06173.ADA	EF06174.ADA	EF06175.ADA	EF06176.ADA	EF06177.ADA	EF06178.ADA
EF06179.ADA	EF06180.ADA	EF06182.ADA	EF06183.ADA	EF06184.ADA	EF06185.ADA
EF06186.ADA	EF06187.ADA	EF06188.ADA	EF06189.ADA	EF06190.ADA	EF06191.ADA
EF06192.ADA	EF06193.ADA	EF06194.ADA	EF06195.ADA	EF06196.ADA	EF06197.ADA
EF06198.ADA	EF06199.ADA	EF0619A.ADA	EF0619B.ADA	EF0619C.ADA	EF0619D.ADA
EF0619E.ADA	EF0619F.ADA	EF0619G.ADA	EF0619H.ADA	EF0619I.ADA	EF0619J.ADA
EF0619K.ADA	EF0619L.ADA	EF0619M.ADA	EF0619N.ADA	EF0619O.ADA	EF0619P.ADA
EF0619Q.ADA	EF0619R.ADA	EF06422.ADA	EF06423.ADA	EF06424.ADA	EF06425.ADA
EF06426.ADA	EF06427.ADA	EF06428.ADA	EF06429.ADA	EF0642A.ADA	EF0642B.ADA
EF06802.ADA	EF06803.ADA	EF06806.ADA	EF06807.ADA	EF06808.ADA	EF06809.ADA
EF0680A.ADA	EF0680B.ADA	EF0680C.ADA	EF0680D.ADA	EF0680E.ADA	EF0680F.ADA
EF06810.ADA	EF06811.ADA	EF06814.ADA	EF06815.ADA	EF06816.ADA	EF06817.ADA
EF06818.ADA	EF06819.ADA	EF06820.ADA	EF06821.ADA	EF06822.ADA	EF06823.ADA
EF06824.ADA	EF06825.ADA	EF09500.ADA	EF09501.ADA	EF09502.ADA	EF09503.ADA
EF09504.ADA	EF09505.ADA	EF09506.ADA	EF09507.ADA	EF09508.ADA	EF09509.ADA
EF09600.ADA	EF09601.ADA	EF09602.ADA	EF09603.ADA	EF09604.ADA	EF09605.ADA
EF09606.ADA	EF09607.ADA	EF09710.ADA	EF09711.ADA	EF09712.ADA	EF09713.ADA
EF09720.ADA	EF09721.ADA	EF09731.ADA	EF09900.ADA	EF09901.ADA	EF09902.ADA
EF09903.ADA	EF09B01.ADA	EF0C100.ADA	EF0C101.ADA	EF0C102.ADA	EF0C103.ADA
EF0C300.ADA	EF0C301.ADA	EF0C302.ADA	EF0C303.ADA	EF0C304.ADA	EF0C305.ADA
EF0C306.ADA	EF0C307.ADA	EF0P000.ADA	EFD3551.ADA	EFD412A.ADA	EFD412C.ADA
EFD412D.ADA	EFD4135.ADA	EFD4136.ADA	EFD4137.ADA	EFD4138.ADA	EFD4139.ADA
EFD413A.ADA	EFD413B.ADA	EFD413C.ADA	EFD413E.ADA	EFD413G.ADA	EFD451K.ADA
EFD451L.ADA	EFD451M.ADA	EFD451N.ADA	EFD451O.ADA	EFD451P.ADA	EFD451Q.ADA
EFD451R.ADA	EFD451S.ADA	EFD4523.ADA	EFD4524.ADA	EFD4525.ADA	EFD4526.ADA
EFD4528.ADA	EFD452A.ADA	EFD455D.ADA	EFD455E.ADA	EFD455F.ADA	EFD455G.ADA
EFD455I.ADA	EFD455J.ADA	EFD455K.ADA	EFD455L.ADA	EFD4566.ADA	EFD4567.ADA
EFD4568.ADA	EFD4569.ADA	EFD4603.ADA	EFD4604.ADA	EFD4605.ADA	EFD4606.ADA
EFD4607.ADA	EFD4608.ADA	EFD4609.ADA	EFD460A.ADA	EFD460B.ADA	EFD460K.ADA
EFD460L.ADA	EFD460N.ADA	EFD4600.ADA	EFD520C.ADA	EFD520D.ADA	EFD520E.ADA
EFD520F.ADA	EFD520G.ADA	EFD520H.ADA	EFD520I.ADA	EFD520J.ADA	EFD520N.ADA
EFD520P.ADA	EFD520Q.ADA	EFD5218.ADA	EFD5219.ADA	EFD521A.ADA	EFD521B.ADA
EFD521C.ADA	EFD521D.ADA	EFD521E.ADA	EFD521F.ADA	EFD521K.ADA	EFD521M.ADA
EFD5210.ADA	EFD521P.ADA	EFD6201.ADA	EFD6209.ADA	EFD6210.ADA	EFD6211.ADA
EFD6213.ADA	EFD6214.ADA	EFD6215.ADA	EFD6216.ADA	EFD6217.ADA	EFD6218.ADA
EFD6219.ADA	EFD6222.ADA	EFD6223.ADA	EFD6224.ADA	EFD6225.ADA	EFD6226.ADA
EFD6227.ADA	EFD6228.ADA	EFD6229.ADA	EFD6230.ADA	EFD6231.ADA	EFD6232.ADA
EFD6233.ADA	EFD6243.ADA	EFD6244.ADA	EFD6245.ADA	EFD6246.ADA	EFD6247.ADA
EFD6248.ADA	EFD6249.ADA	EFD6250.ADA	EFD6251.ADA	EFD6252.ADA	EFD6253.ADA
EFD6260.ADA	EFD6261.ADA	EFD6262.ADA	EFD6263.ADA	EFD6264.ADA	EFD6265.ADA
EFD6266.ADA	EFD6267.ADA	EFD6268.ADA	EFD6269.ADA	EFD6270.ADA	EFD6271.ADA
EFD6272.ADA	EFD6273.ADA	EFD6274.ADA	EFD6275.ADA	EFD6276.ADA	EFD6277.ADA
EFD6278.ADA	EFD6279.ADA	EFD6300.ADA	EFD6301.ADA	EFD6308.ADA	EFD6309.ADA
EFD6310.ADA	EFD6311.ADA	EFD6312.ADA	EFD6313.ADA	EFD6314.ADA	EFD6315.ADA
EFD6316.ADA	EFD6317.ADA	EFD6318.ADA	EFD6319.ADA	EFD6322.ADA	EFD6332.ADA
EFD6333.ADA	EFD6334.ADA	EFD6335.ADA	EFD6336.ADA	EFD6337.ADA	EFD6338.ADA
EFD6339.ADA	EFD6340.ADA	EFD6342.ADA	EFD6343.ADA	EFD6344.ADA	EFD6345.ADA
EFD6346.ADA	EFD6347.ADA	EFD6348.ADA	EFD6349.ADA	EFD6350.ADA	EFD6352.ADA
EFD6353.ADA	EFD6354.ADA	EFD6355.ADA	EFD6356.ADA	EFD6357.ADA	EFD6358.ADA
EFD6359.ADA	EFD6360.ADA	EFD6362.ADA	EFD6363.ADA	EFD6364.ADA	EFD6365.ADA
EFD6366.ADA	EFD6367.ADA	EFD6368.ADA	EFD6369.ADA	EFD6370.ADA	EFD6372.ADA
EFD6373.ADA	EFD6374.ADA	EFD6375.ADA	EFD6376.ADA	EFD6377.ADA	EFD6378.ADA
EFD6379.ADA	EFD6380.ADA	EFD6382.ADA	EFD6383.ADA	EFD6384.ADA	EFD6385.ADA
EFD6386.ADA	EFD6387.ADA	EFD6388.ADA	EFD6389.ADA	EFD6390.ADA	EFD680G.ADA
EFD680H.ADA	EFD680I.ADA	EFD680J.ADA	EFD9000.ADA	EFD9200.ADA	EFD9C00.ADA
EFD9C01.ADA	EFD9C02.ADA	EFDC104.ADA	EFDC105.ADA	EFDC106.ADA	EFDC107.ADA
EFDC310.ADA	EFDC311.ADA	EFDC313.ADA	EFDC314.ADA	EFDD600.ADA	EFDD601.ADA
EFDD602.ADA	EFDD603.ADA	EFDD604.ADA	EFDD605.ADA	EFDD606.ADA	EFDD607.ADA
EFDD609.ADA	EFDD610.ADA	EFDD611.ADA	EFDDA01.ADA	EFDDA02.ADA	EFDE000.ADA
EFDE201.ADA	EFDE202.ADA	EFDE203.ADA	EFDE204.ADA	EFDE205.ADA	EFDE206.ADA
EFDE207.ADA	EFDE208.ADA	EFDE209.ADA	EFDE20A.ADA	EFDE211.ADA	EFDE212.ADA
EFDE213.ADA	EFDE214.ADA	EFDE215.ADA	EFDE216.ADA	EFDE220.ADA	EFDE231.ADA
EFDE232.ADA	EFDE233.ADA	EFDE234.ADA	EFDE235.ADA	EFDE236.ADA	EFDE237.ADA
EFDE238.ADA	EFDE239.ADA	EFDE23A.ADA	EFDE241.ADA	EFDE242.ADA	EFDE243.ADA
EFDE244.ADA	EFDE245.ADA	EFDE246.ADA	EFDE250.ADA	EFDE401.ADA	EFDE402.ADA
EFDE403.ADA	EFDE404.ADA	EFDE405.ADA	EFDE406.ADA	EFDE407.ADA	EFDE408.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

EFDE409.ADA	EFDE40A.ADA	EFDE411.ADA	EFDE412.ADA	EFDE413.ADA	EFDE414.ADA
EFDE415.ADA	EFDE416.ADA	EFDE420.ADA	EFDE431.ADA	EFDE432.ADA	EFDE433.ADA
EFDE434.ADA	EFDE435.ADA	EFDE436.ADA	EFDE437.ADA	EFDE438.ADA	EFDE439.ADA
EFDE43A.ADA	EFDE441.ADA	EFDE442.ADA	EFDE443.ADA	EFDE444.ADA	EFDE445.ADA
EFDE446.ADA	EFDE450.ADA	EFDF000.ADA	EFDF001.ADA	EFDF002.ADA	EFDF003.ADA
EFDF004.ADA	EFDF005.ADA	EFM9A00.ADA	EFM9A01.ADA	EFM9A02.ADA	EFM9A03.ADA
EFN9300.ADA	EFN9301.ADA	EFN9302.ADA	EFN9303.ADA	EFN9510.ADA	EFN9511.ADA
EFN9610.ADA	EFN9611.ADA	EFN9612.ADA	EFN9613.ADA	EG0000C.ADA	EG0P000.ADA
EGD0000.ADA	EGD0001.ADA	EGD0002.ADA	EGD0003.ADA	EGD0004.ADA	EGD0005.ADA
EGD0006.ADA	EGD0007.ADA	EGD0008.ADA	EGD0009.ADA	EGD000D.ADA	EGD000E.ADA
EGD0019.ADA	EGD0029.ADA	EGD0039.ADA	EGD0049.ADA	EGN000F.ADA	EGN000G.ADA
EGN000H.ADA	EGN000I.ADA	EGN000J.ADA	EGN000K.ADA	EL09000.ADA	EL09100.ADA
EL09101.ADA	EL09111.ADA	EL09121.ADA	EL09131.ADA	EL0P000.ADA	ELD9200.ADA
ELD9202.ADA	ELD9203.ADA	ELD9212.ADA	ELD9213.ADA	ELD9222.ADA	ELD9223.ADA
ELD9232.ADA	ELD9233.ADA	ELDE000.ADA	ELDE2B1.ADA	ELDE2B2.ADA	ELDE2C1.ADA
ELDE2C2.ADA	ELDE2C5.ADA	ELDE2C6.ADA	ELDE4B1.ADA	ELDE4B2.ADA	ELDE4C1.ADA
ELDE4C2.ADA	ELDE4C5.ADA	ELDE4C6.ADA	E000001.ADA	E000002.ADA	E000102.ADA
E000103.ADA	E000104.ADA	E000107.ADA	E000108.ADA	E000109.ADA	E000112.ADA
E000113.ADA	E000114.ADA	E000202.ADA	E000203.ADA	E000204.ADA	E000207.ADA
E000208.ADA	E000209.ADA	E000300.ADA	E000305.ADA	E000310.ADA	E000311.ADA
E000312.ADA	E000313.ADA	E000314.ADA	E000315.ADA	E000316.ADA	E000317.ADA
E000318.ADA	E000319.ADA	E000320.ADA	E000321.ADA	E000322.ADA	E000323.ADA
E000324.ADA	E000325.ADA	E000326.ADA	E000327.ADA	E000328.ADA	E000329.ADA
E000330.ADA	E000331.ADA	E000332.ADA	E000333.ADA	E000334.ADA	E000335.ADA
E000336.ADA	E000337.ADA	E000338.ADA	E000339.ADA	E000340.ADA	E000341.ADA
E000342.ADA	E000343.ADA	E000347.ADA	E000348.ADA	E000349.ADA	E000350.ADA
E000351.ADA	E000352.ADA	E000353.ADA	E000354.ADA	E000355.ADA	E000356.ADA
E000357.ADA	E000358.ADA	E000359.ADA	E000360.ADA	E000361.ADA	E000362.ADA
E000363.ADA	E000364.ADA	E000365.ADA	E000366.ADA	E000367.ADA	E000368.ADA
E000369.ADA	E00036A.ADA	E00036B.ADA	E00036F.ADA	E00036G.ADA	E00036H.ADA
E00036I.ADA	E00036J.ADA	E00036K.ADA	E00036L.ADA	E00036M.ADA	E00036N.ADA
E00036U.ADA	E00036V.ADA	E00036W.ADA	E000374.ADA	E000375.ADA	E000376.ADA
E000377.ADA	E000378.ADA	E000390.ADA	E000391.ADA	E000402.ADA	E000403.ADA
E000404.ADA	E000500.ADA	E000502.ADA	E000503.ADA	E000504.ADA	E000507.ADA
E000508.ADA	E000509.ADA	E000512.ADA	E000513.ADA	E000514.ADA	E000517.ADA
E000518.ADA	E000519.ADA	E000602.ADA	E000603.ADA	E000604.ADA	E000605.ADA
E000606.ADA	E000607.ADA	E000702.ADA	E000703.ADA	E000704.ADA	E000705.ADA
E000706.ADA	E000707.ADA	E000708.ADA	E000709.ADA	E000710.ADA	E000711.ADA
E000713.ADA	E000714.ADA	E000715.ADA	E000716.ADA	E000717.ADA	E000718.ADA
E000719.ADA	E000720.ADA	E000721.ADA	E00P000.ADA	E0D0344.ADA	E0D0345.ADA
E0D0346.ADA	E0D036C.ADA	E0D036D.ADA	E0D036E.ADA	E0D0360.ADA	E0D036P.ADA
E0D036Q.ADA	E0D036R.ADA	E0D036S.ADA	E0D036T.ADA	E0D036X.ADA	E0D036Y.ADA
E0D036Z.ADA	E0D0370.ADA	E0D0371.ADA	E0D0372.ADA	E0D0373.ADA	E0D0380.ADA
E0D0381.ADA	E0D0382.ADA	E0D0383.ADA	E0D0523.ADA	E0D0524.ADA	E0D0525.ADA
E0D0526.ADA	E0D0527.ADA	E0D0528.ADA	E0D0529.ADA	OURDMP.ADA	OURSYS.ADA
OURSYSR.ADA	SA00000.ADA	SA0P000.ADA	SF03500.ADA	SF03501.ADA	SF03502.ADA
SF03503.ADA	SF03504.ADA	SF03509.ADA	SF03510.ADA	SF03511.ADA	SF03512.ADA
SF03513.ADA	SF03514.ADA	SF03517.ADA	SF03518.ADA	SF03519.ADA	SF03550.ADA
SF03600.ADA	SF03601.ADA	SF03602.ADA	SF03603.ADA	SF03604.ADA	SF03605.ADA
SF03606.ADA	SF03607.ADA	SF03608.ADA	SF03609.ADA	SF03610.ADA	SF03611.ADA
SF03612.ADA	SF03613.ADA	SF03614.ADA	SF03615.ADA	SF03616.ADA	SF03617.ADA
SF03618.ADA	SF03619.ADA	SF03620.ADA	SF03630.ADA	SF03631.ADA	SF03632.ADA
SF03633.ADA	SF03634.ADA	SF03635.ADA	SF03636.ADA	SF03637.ADA	SF03638.ADA
SF03639.ADA	SF03641.ADA	SF03642.ADA	SF03643.ADA	SF03644.ADA	SF03645.ADA
SF03646.ADA	SF03647.ADA	SF03648.ADA	SF03649.ADA	SF03650.ADA	SF03700.ADA
SF03701.ADA	SF03702.ADA	SF03703.ADA	SF03704.ADA	SF03800.ADA	SF03801.ADA
SF03802.ADA	SF03803.ADA	SF03804.ADA	SF03805.ADA	SF04120.ADA	SF04121.ADA
SF04122.ADA	SF04123.ADA	SF04124.ADA	SF04125.ADA	SF04126.ADA	SF04127.ADA
SF04129.ADA	SF0412B.ADA	SF04130.ADA	SF04131.ADA	SF04132.ADA	SF04133.ADA
SF0413D.ADA	SF0413H.ADA	SF0413I.ADA	SF0413J.ADA	SF0413K.ADA	SF04310.ADA
SF04311.ADA	SF04312.ADA	SF04510.ADA	SF04511.ADA	SF04512.ADA	SF04513.ADA
SF04514.ADA	SF04515.ADA	SF04516.ADA	SF04517.ADA	SF04518.ADA	SF04519.ADA
SF0451A.ADA	SF0451B.ADA	SF0451C.ADA	SF0451D.ADA	SF0451E.ADA	SF0451F.ADA
SF0451G.ADA	SF0451H.ADA	SF0451I.ADA	SF0451J.ADA	SF0451V.ADA	SF0451W.ADA
SF0451X.ADA	SF0451Y.ADA	SF0451Z.ADA	SF04520.ADA	SF04521.ADA	SF04522.ADA
SF04527.ADA	SF04529.ADA	SF04530.ADA	SF04531.ADA	SF04532.ADA	SF04533.ADA
SF04534.ADA	SF04535.ADA	SF04536.ADA	SF04537.ADA	SF04538.ADA	SF04539.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)[illegible]

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

SFD521E.ADA	SFD521F.ADA	SFD521K.ADA	SFD521M.ADA	SFD521O.ADA	SFD521P.ADA
SFD6201.ADA	SFD6209.ADA	SFD6210.ADA	SFD6211.ADA	SFD6213.ADA	SFD6214.ADA
SFD6215.ADA	SFD6216.ADA	SFD6217.ADA	SFD6218.ADA	SFD6219.ADA	SFD6222.ADA
SFD6223.ADA	SFD6224.ADA	SFD6225.ADA	SFD6226.ADA	SFD6227.ADA	SFD6228.ADA
SFD6229.ADA	SFD6230.ADA	SFD6231.ADA	SFD6232.ADA	SFD6233.ADA	SFD6243.ADA
SFD6244.ADA	SFD6245.ADA	SFD6246.ADA	SFD6247.ADA	SFD6248.ADA	SFD6249.ADA
SFD6250.ADA	SFD6251.ADA	SFD6252.ADA	SFD6253.ADA	SFD6260.ADA	SFD6261.ADA
SFD6262.ADA	SFD6263.ADA	SFD6264.ADA	SFD6265.ADA	SFD6266.ADA	SFD6267.ADA
SFD6268.ADA	SFD6269.ADA	SFD6270.ADA	SFD6271.ADA	SFD6272.ADA	SFD6273.ADA
SFD6274.ADA	SFD6275.ADA	SFD6276.ADA	SFD6277.ADA	SFD6278.ADA	SFD6279.ADA
SFD6300.ADA	SFD6301.ADA	SFD6308.ADA	SFD6309.ADA	SFD6310.ADA	SFD6311.ADA
SFD6312.ADA	SFD6313.ADA	SFD6314.ADA	SFD6315.ADA	SFD6316.ADA	SFD6317.ADA
SFD6318.ADA	SFD6319.ADA	SFD6322.ADA	SFD6332.ADA	SFD6333.ADA	SFD6334.ADA
SFD6335.ADA	SFD6336.ADA	SFD6337.ADA	SFD6338.ADA	SFD6339.ADA	SFD6340.ADA
SFD6342.ADA	SFD6343.ADA	SFD6344.ADA	SFD6345.ADA	SFD6346.ADA	SFD6347.ADA
SFD6348.ADA	SFD6349.ADA	SFD6350.ADA	SFD6352.ADA	SFD6353.ADA	SFD6354.ADA
SFD6355.ADA	SFD6356.ADA	SFD6357.ADA	SFD6358.ADA	SFD6359.ADA	SFD6360.ADA
SFD6362.ADA	SFD6363.ADA	SFD6364.ADA	SFD6365.ADA	SFD6366.ADA	SFD6367.ADA
SFD6368.ADA	SFD6369.ADA	SFD6370.ADA	SFD6372.ADA	SFD6373.ADA	SFD6374.ADA
SFD6375.ADA	SFD6376.ADA	SFD6377.ADA	SFD6378.ADA	SFD6379.ADA	SFD6380.ADA
SFD6382.ADA	SFD6383.ADA	SFD6384.ADA	SFD6385.ADA	SFD6386.ADA	SFD6387.ADA
SFD6388.ADA	SFD6389.ADA	SFD6390.ADA	SFD680G.ADA	SFD680H.ADA	SFD680I.ADA
SFD680J.ADA	SFD9000.ADA	SFD9200.ADA	SFD9C00.ADA	SFD9C01.ADA	SFD9C02.ADA
SFDC104.ADA	SFDC105.ADA	SFDC106.ADA	SFDC107.ADA	SFDC310.ADA	SFDC311.ADA
SFDC313.ADA	SFDC314.ADA	SFDD600.ADA	SFDD601.ADA	SFDD602.ADA	SFDD603.ADA
SFDD604.ADA	SFDD605.ADA	SFDD606.ADA	SFDD607.ADA	SFDD609.ADA	SFDD610.ADA
SFDD611.ADA	SFDD722.ADA	SFDD723.ADA	SFDD724.ADA	SFDD725.ADA	SFDD72A.ADA
SFDD72B.ADA	SFDD72C.ADA	SFDD72E.ADA	SFDD72F.ADA	SFDD72G.ADA	SFDD72H.ADA
SFDD72I.ADA	SFDDA01.ADA	SFDDA02.ADA	SFDE000.ADA	SFDE201.ADA	SFDE202.ADA
SFDE203.ADA	SFDE204.ADA	SFDE205.ADA	SFDE206.ADA	SFDE207.ADA	SFDE208.ADA
SFDE209.ADA	SFDE20A.ADA	SFDE211.ADA	SFDE212.ADA	SFDE213.ADA	SFDE214.ADA
SFDE215.ADA	SFDE216.ADA	SFDE220.ADA	SFDE231.ADA	SFDE232.ADA	SFDE233.ADA
SFDE234.ADA	SFDE235.ADA	SFDE236.ADA	SFDE237.ADA	SFDE238.ADA	SFDE239.ADA
SFDE23A.ADA	SFDE241.ADA	SFDE242.ADA	SFDE243.ADA	SFDE244.ADA	SFDE245.ADA
SFDE246.ADA	SFDE250.ADA	SFDE401.ADA	SFDE402.ADA	SFDE403.ADA	SFDE404.ADA
SFDE405.ADA	SFDE406.ADA	SFDE407.ADA	SFDE408.ADA	SFDE409.ADA	SFDE40A.ADA
SFDE411.ADA	SFDE412.ADA	SFDE413.ADA	SFDE414.ADA	SFDE415.ADA	SFDE416.ADA
SFDE420.ADA	SFDE431.ADA	SFDE432.ADA	SFDE433.ADA	SFDE434.ADA	SFDE435.ADA
SFDE436.ADA	SFDE437.ADA	SFDE438.ADA	SFDE439.ADA	SFDE43A.ADA	SFDE441.ADA
SFDE442.ADA	SFDE443.ADA	SFDE444.ADA	SFDE445.ADA	SFDE446.ADA	SFDE450.ADA
SFDF000.ADA	SFDF001.ADA	SFDF002.ADA	SFDF003.ADA	SFDF004.ADA	SFDF005.ADA
SFM9A00.ADA	SFM9A01.ADA	SFM9A02.ADA	SFM9A03.ADA	SFMB000.ADA	SFMB001.ADA
SFMB002.ADA	SFMB003.ADA	SFMB004.ADA	SFMB005.ADA	SFMB006.ADA	SFMB007.ADA
SFMB008.ADA	SFMB009.ADA	SFMB010.ADA	SFMB011.ADA	SFMB012.ADA	SFMB013.ADA
SFMB014.ADA	SFMB015.ADA	SFMB016.ADA	SFMB017.ADA	SFMB018.ADA	SFMB019.ADA
SFMB020.ADA	SFMB021.ADA	SFMB022.ADA	SFMB023.ADA	SFMB024.ADA	SFMB025.ADA
SFMB026.ADA	SFMB027.ADA	SFMB028.ADA	SFMB029.ADA	SFMB030.ADA	SFMB031.ADA
SFMB032.ADA	SFMB033.ADA	SFMB034.ADA	SFMB035.ADA	SFMB036.ADA	SFMB037.ADA
SFMB038.ADA	SFMB039.ADA	SFMB040.ADA	SFMB041.ADA	SFMB042.ADA	SFMB043.ADA
SFMB044.ADA	SFN9300.ADA	SFN9301.ADA	SFN9302.ADA	SFN9303.ADA	SFN9510.ADA
SFN9511.ADA	SFN9610.ADA	SFN9611.ADA	SFN9612.ADA	SFN9613.ADA	SG0000C.ADA
SG0P000.ADA	SGD0000.ADA	SGD0001.ADA	SGD0002.ADA	SGD0003.ADA	SGD0004.ADA
SGD0005.ADA	SGD0006.ADA	SGD0007.ADA	SGD0008.ADA	SGD0009.ADA	SGD000D.ADA
SGD000E.ADA	SGD0019.ADA	SGD0029.ADA	SGD0039.ADA	SGD0049.ADA	SGN000F.ADA
SGN000G.ADA	SGN000H.ADA	SGN000I.ADA	SGN000J.ADA	SGN000K.ADA	SL09000.ADA
SL09100.ADA	SL09101.ADA	SL09111.ADA	SL09121.ADA	SL09131.ADA	SL0P000.ADA
SLD9200.ADA	SLD9202.ADA	SLD9203.ADA	SLD9212.ADA	SLD9213.ADA	SLD9222.ADA
SLD9223.ADA	SLD9232.ADA	SLD9233.ADA	SLDE000.ADA	SLDE2B1.ADA	SLDE2B2.ADA
SLDE2C1.ADA	SLDE2C2.ADA	SLDE2C5.ADA	SLDE2C6.ADA	SLDE4B1.ADA	SLDE4B2.ADA
SLDE4C1.ADA	SLDE4C2.ADA	SLDE4C5.ADA	SLDE4C6.ADA	S000001.ADA	S000002.ADA
S000102.ADA	S000103.ADA	S000104.ADA	S000107.ADA	S000108.ADA	S000109.ADA
S000112.ADA	S000113.ADA	S000114.ADA	S000202.ADA	S000203.ADA	S000204.ADA
S000207.ADA	S000208.ADA	S000209.ADA	S000300.ADA	S000305.ADA	S000310.ADA
S000311.ADA	S000312.ADA	S000313.ADA	S000314.ADA	S000315.ADA	S000316.ADA
S000317.ADA	S000318.ADA	S000319.ADA	S000320.ADA	S000321.ADA	S000322.ADA
S000323.ADA	S000324.ADA	S000325.ADA	S000326.ADA	S000327.ADA	S000328.ADA
S000329.ADA	S000330.ADA	S000331.ADA	S000332.ADA	S000333.ADA	S000334.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

S000335.ADA	S000336.ADA	S000337.ADA	S000338.ADA	S000339.ADA	S000340.ADA
S000341.ADA	S000342.ADA	S000343.ADA	S000347.ADA	S000348.ADA	S000349.ADA
S000350.ADA	S000351.ADA	S000352.ADA	S000353.ADA	S000354.ADA	S000355.ADA
S000356.ADA	S000357.ADA	S000358.ADA	S000359.ADA	S000360.ADA	S000361.ADA
S000362.ADA	S000363.ADA	S000364.ADA	S000365.ADA	S000366.ADA	S000367.ADA
S000368.ADA	S000369.ADA	S00036A.ADA	S00036B.ADA	S00036F.ADA	S00036G.ADA
S00036H.ADA	S00036I.ADA	S00036J.ADA	S00036K.ADA	S00036L.ADA	S00036M.ADA
S00036N.ADA	S00036U.ADA	S00036V.ADA	S00036W.ADA	S000374.ADA	S000375.ADA
S000376.ADA	S000377.ADA	S000378.ADA	S000390.ADA	S000391.ADA	S000402.ADA
S000403.ADA	S000404.ADA	S000500.ADA	S000502.ADA	S000503.ADA	S000504.ADA
S000507.ADA	S000508.ADA	S000509.ADA	S000512.ADA	S000513.ADA	S000514.ADA
S000517.ADA	S000518.ADA	S000519.ADA	S000602.ADA	S000603.ADA	S000604.ADA
S000605.ADA	S000606.ADA	S000607.ADA	S000702.ADA	S000703.ADA	S000704.ADA
S000705.ADA	S000706.ADA	S000707.ADA	S000708.ADA	S000709.ADA	S000710.ADA
S000711.ADA	S000713.ADA	S000714.ADA	S000715.ADA	S000716.ADA	S000717.ADA
S000718.ADA	S000719.ADA	S000720.ADA	S000721.ADA	S00P000.ADA	S0D0344.ADA
S0D0345.ADA	S0D0346.ADA	S0D036C.ADA	S0D036D.ADA	S0D036E.ADA	S0D0360.ADA
S0D036P.ADA	S0D036Q.ADA	S0D036R.ADA	S0D036S.ADA	S0D036T.ADA	S0D036X.ADA
S0D036Y.ADA	S0D036Z.ADA	S0D0370.ADA	S0D0371.ADA	S0D0372.ADA	S0D0373.ADA
S0D0380.ADA	S0D0381.ADA	S0D0382.ADA	S0D0383.ADA	S0D0523.ADA	S0D0524.ADA
S0D0525.ADA	S0D0526.ADA	S0D0527.ADA	S0D0528.ADA	S0D0529.ADA	TA00000.ADA
TA0P000.ADA	TF03500.ADA	TF03501.ADA	TF03502.ADA	TF03503.ADA	TF03504.ADA
TF03509.ADA	TF03510.ADA	TF03511.ADA	TF03512.ADA	TF03513.ADA	TF03514.ADA
TF03517.ADA	TF03518.ADA	TF03519.ADA	TF03550.ADA	TF03600.ADA	TF03601.ADA
TF03602.ADA	TF03603.ADA	TF03604.ADA	TF03605.ADA	TF03606.ADA	TF03607.ADA
TF03608.ADA	TF03609.ADA	TF03610.ADA	TF03611.ADA	TF03612.ADA	TF03613.ADA
TF03614.ADA	TF03615.ADA	TF03616.ADA	TF03617.ADA	TF03618.ADA	TF03619.ADA
TF03620.ADA	TF03630.ADA	TF03631.ADA	TF03632.ADA	TF03633.ADA	TF03634.ADA
TF03635.ADA	TF03636.ADA	TF03637.ADA	TF03638.ADA	TF03639.ADA	TF03641.ADA
TF03642.ADA	TF03643.ADA	TF03644.ADA	TF03645.ADA	TF03646.ADA	TF03647.ADA
TF03648.ADA	TF03649.ADA	TF03650.ADA	TF03700.ADA	TF03701.ADA	TF03702.ADA
TF03703.ADA	TF03704.ADA	TF03800.ADA	TF03801.ADA	TF03802.ADA	TF03803.ADA
TF03804.ADA	TF03805.ADA	TF04120.ADA	TF04121.ADA	TF04122.ADA	TF04123.ADA
TF04124.ADA	TF04125.ADA	TF04126.ADA	TF04127.ADA	TF04129.ADA	TF0412B.ADA
TF04130.ADA	TF04131.ADA	TF04132.ADA	TF04133.ADA	TF0413D.ADA	TF0413H.ADA
TF04131.ADA	TF0413J.ADA	TF0413K.ADA	TF04310.ADA	TF04311.ADA	TF04312.ADA
TF04510.ADA	TF04511.ADA	TF04512.ADA	TF04513.ADA	TF04514.ADA	TF04515.ADA
TF04516.ADA	TF04517.ADA	TF04518.ADA	TF04519.ADA	TF0451A.ADA	TF0451B.ADA
TF0451C.ADA	TF0451D.ADA	TF0451E.ADA	TF0451F.ADA	TF0451G.ADA	TF0451H.ADA
TF0451I.ADA	TF0451J.ADA	TF0451V.ADA	TF0451W.ADA	TF0451X.ADA	TF0451Y.ADA
TF0451Z.ADA	TF04520.ADA	TF04521.ADA	TF04522.ADA	TF04527.ADA	TF04529.ADA
TF04530.ADA	TF04531.ADA	TF04532.ADA	TF04533.ADA	TF04534.ADA	TF04535.ADA
TF04536.ADA	TF04537.ADA	TF04538.ADA	TF04539.ADA	TF0453A.ADA	TF0453B.ADA
TF0453C.ADA	TF04540.ADA	TF04541.ADA	TF04550.ADA	TF04551.ADA	TF04552.ADA
TF04553.ADA	TF04554.ADA	TF04555.ADA	TF04556.ADA	TF04557.ADA	TF04558.ADA
TF04559.ADA	TF0455N.ADA	TF04550.ADA	TF04560.ADA	TF04562.ADA	TF04563.ADA
TF04600.ADA	TF04601.ADA	TF04602.ADA	TF0460E.ADA	TF0460F.ADA	TF0460G.ADA
TF0460H.ADA	TF0460I.ADA	TF0460J.ADA	TF0460M.ADA	TF05200.ADA	TF05201.ADA
TF05202.ADA	TF05203.ADA	TF05204.ADA	TF05205.ADA	TF05206.ADA	TF05207.ADA
TF05208.ADA	TF05209.ADA	TF0520A.ADA	TF0520B.ADA	TF0520M.ADA	TF05200.A

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

TF06101.ADA	TF06108.ADA	TF06109.ADA	TF06110.ADA	TF06111.ADA	TF06112.ADA
TF06113.ADA	TF06114.ADA	TF06115.ADA	TF06116.ADA	TF06117.ADA	TF06118.ADA
TF06119.ADA	TF06122.ADA	TF06132.ADA	TF06133.ADA	TF06134.ADA	TF06135.ADA
TF06136.ADA	TF06137.ADA	TF06138.ADA	TF06139.ADA	TF06140.ADA	TF06142.ADA
TF06143.ADA	TF06144.ADA	TF06145.ADA	TF06146.ADA	TF06147.ADA	TF06148.ADA
TF06149.ADA	TF06150.ADA	TF06152.ADA	TF06153.ADA	TF06154.ADA	TF06155.ADA
TF06156.ADA	TF06157.ADA	TF06158.ADA	TF06159.ADA	TF06160.ADA	TF06162.ADA
TF06163.ADA	TF06164.ADA	TF06165.ADA	TF06166.ADA	TF06167.ADA	TF06168.ADA
TF06169.ADA	TF06170.ADA	TF06172.ADA	TF06173.ADA	TF06174.ADA	TF06175.ADA
TF06176.ADA	TF06177.ADA	TF06178.ADA	TF06179.ADA	TF06180.ADA	TF06182.ADA
TF06183.ADA	TF06184.ADA	TF06185.ADA	TF06186.ADA	TF06187.ADA	TF06188.ADA
TF06189.ADA	TF06190.ADA	TF06191.ADA	TF06192.ADA	TF06193.ADA	TF06194.ADA
TF06195.ADA	TF06196.ADA	TF06197.ADA	TF06198.ADA	TF06199.ADA	TF0619A.ADA
TF0619B.ADA	TF0619C.ADA	TF0619D.ADA	TF0619E.ADA	TF0619F.ADA	TF0619G.ADA
TF0619H.ADA	TF0619I.ADA	TF0619J.ADA	TF0619K.ADA	TF0619L.ADA	TF0619M.ADA
TF0619N.ADA	TF0619O.ADA	TF0619P.ADA	TF0619Q.ADA	TF0619R.ADA	TF06422.ADA
TF06423.ADA	TF06424.ADA	TF06425.ADA	TF06426.ADA	TF06427.ADA	TF06428.ADA
TF06429.ADA	TF0642A.ADA	TF0642B.ADA	TF06802.ADA	TF06803.ADA	TF06806.ADA
TF06807.ADA	TF06808.ADA	TF06809.ADA	TF0680A.ADA	TF0680B.ADA	TF0680C.ADA
TF0680D.ADA	TF0680E.ADA	TF0680F.ADA	TF06810.ADA	TF06811.ADA	TF06814.ADA
TF06815.ADA	TF06816.ADA	TF06817.ADA	TF06818.ADA	TF06819.ADA	TF06820.ADA
TF06821.ADA	TF06822.ADA	TF06823.ADA	TF06824.ADA	TF06825.ADA	TF09500.ADA
TF09501.ADA	TF09502.ADA	TF09503.ADA	TF09504.ADA	TF09505.ADA	TF09506.ADA
TF09507.ADA	TF09508.ADA	TF09509.ADA	TF09600.ADA	TF09601.ADA	TF09602.ADA
TF09603.ADA	TF09604.ADA	TF09605.ADA	TF09606.ADA	TF09607.ADA	TF09710.ADA
TF09711.ADA	TF09712.ADA	TF09713.ADA	TF09720.ADA	TF09721.ADA	TF09731.ADA
TF09900.ADA	TF09901.ADA	TF09902.ADA	TF09903.ADA	TF09B01.ADA	TF0C100.ADA
TF0C101.ADA	TF0C102.ADA	TF0C103.ADA	TF0C300.ADA	TF0C301.ADA	TF0C302.ADA
TF0C303.ADA	TF0C304.ADA	TF0C305.ADA	TF0C306.ADA	TF0C307.ADA	TF0D720.ADA
TF0D721.ADA	TF0D727.ADA	TF0D728.ADA	TF0D729.ADA	TF0D72D.ADA	TF0P000.ADA
TFD3551.ADA	TFD412A.ADA	TFD412C.ADA	TFD412D.ADA	TFD4135.ADA	TFD4136.ADA
TFD4137.ADA	TFD4138.ADA	TFD4139.ADA	TFD413A.ADA	TFD413B.ADA	TFD413C.ADA
TFD413E.ADA	TFD413G.ADA	TFD451K.ADA	TFD451L.ADA	TFD451M.ADA	TFD451N.ADA
TFD451O.ADA	TFD451P.ADA	TFD451Q.ADA	TFD451R.ADA	TFD451S.ADA	TFD4523.ADA
TFD4524.ADA	TFD4525.ADA	TFD4526.ADA	TFD4528.ADA	TFD452A.ADA	TFD455D.ADA
TFD455E.ADA	TFD455F.ADA	TFD455G.ADA	TFD455I.ADA	TFD455J.ADA	TFD455K.ADA
TFD455L.ADA	TFD4567.ADA	TFD4568.ADA	TFD4569.ADA	TFD4569.ADA	TFD4603.ADA
TFD4604.ADA	TFD4605.ADA	TFD4606.ADA	TFD4607.ADA	TFD4608.ADA	TFD4609.ADA
TFD460A.ADA	TFD460B.ADA	TFD460K.ADA	TFD460L.ADA	TFD460N.ADA	TFD4600.ADA
TFD4800.ADA	TFD4801.ADA	TFD4802.ADA	TFD4803.ADA	TFD4804.ADA	TFD520C.ADA
TFD520D.ADA	TFD520E.ADA	TFD520F.ADA	TFD520G.ADA	TFD520H.ADA	TFD520I.ADA
TFD520J.ADA	TFD520N.ADA	TFD520P.ADA	TFD520Q.ADA	TFD5218.ADA	TFD5219.ADA
TFD521A.ADA	TFD521B.ADA	TFD521C.ADA	TFD521D.ADA	TFD521E.ADA	TFD521F.ADA
TFD521K.ADA	TFD521M.ADA	TFD521O.ADA	TFD521P.ADA	TFD6201.ADA	TFD6209.ADA
TFD6210.ADA	TFD6211.ADA	TFD6213.ADA	TFD6214.ADA	TFD6215.ADA	TFD6216.ADA
TFD6217.ADA	TFD6218.ADA	TFD6219.ADA	TFD6222.ADA	TFD6223.ADA	TFD6224.ADA
TFD6225.ADA	TFD6226.ADA	TFD6227.ADA	TFD6228.ADA	TFD6229.ADA	TFD6230.ADA
TFD6231.ADA	TFD6232.ADA	TFD6233.ADA	TFD6243.ADA	TFD6244.ADA	TFD6245.ADA
TFD6246.ADA	TFD6247.ADA	TFD6248.ADA	TFD6249.ADA	TFD6250.ADA	TFD6251.ADA
TFD6252.ADA	TFD6253.ADA	TFD6260.ADA	TFD6261.ADA	TFD6262.ADA	TFD6263.ADA
TFD6264.ADA	TFD6265.ADA	TFD6266.ADA	TFD6267.ADA	TFD6268.ADA	TFD6269.ADA
TFD6270.ADA	TFD6271.ADA	TFD6272.ADA	TFD6273.ADA	TFD6274.ADA	TFD6275.ADA
TFD6276.ADA	TFD6277.ADA	TFD6278.ADA	TFD6279.ADA	TFD6300.ADA	TFD6301.ADA
TFD6308.ADA	TFD6309.ADA	TFD6310.ADA	TFD6311.ADA	TFD6312.ADA	TFD6313.ADA
TFD6314.ADA	TFD6315.ADA	TFD6316.ADA	TFD6317.ADA	TFD6318.ADA	TFD6319.ADA
TFD6322.ADA	TFD6332.ADA	TFD6333.ADA	TFD6334.ADA	TFD6335.ADA	TFD6336.ADA
TFD6337.ADA	TFD6338.ADA	TFD6339.ADA	TFD6340.ADA	TFD6342.ADA	TFD6343.ADA
TFD6344.ADA	TFD6345.ADA	TFD6346.ADA	TFD6347.ADA	TFD6348.ADA	TFD6349.ADA
TFD6350.ADA	TFD6352.ADA	TFD6353.ADA	TFD6354.ADA	TFD6355.ADA	TFD6356.ADA
TFD6357.ADA	TFD6358.ADA	TFD6359.ADA	TFD6360.ADA	TFD6362.ADA	TFD6363.ADA
TFD6364.ADA	TFD6365.ADA	TFD6366.ADA	TFD6367.ADA	TFD6368.ADA	TFD6369.ADA
TFD6370.ADA	TFD6372.ADA	TFD6373.ADA	TFD6374.ADA	TFD6375.ADA	TFD6376.ADA
TFD6377.ADA	TFD6378.ADA	TFD6379.ADA	TFD6380.ADA	TFD6382.ADA	TFD6383.ADA
TFD6384.ADA	TFD6385.ADA	TFD6386.ADA	TFD6387.ADA	TFD6388.ADA	TFD6389.ADA
TFD6390.ADA	TFD680G.ADA	TFD680H.ADA	TFD680I.ADA	TFD680J.ADA	TFD9000.ADA
TFD9200.ADA	TFD9C00.ADA	TFD9C01.ADA	TFD9C02.ADA	TFD9C03.ADA	TFD9C04.ADA
TFD9C106.ADA	TFD9C107.ADA	TFD9C310.ADA	TFD9C311.ADA	TFD9C313.ADA	TFD9C314.ADA

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

TFDD600.ADA	TFDD601.ADA	TFDD602.ADA	TFDD603.ADA	TFDD604.ADA	TFDD605.ADA
TFDD606.ADA	TFDD607.ADA	TFDD609.ADA	TFDD610.ADA	TFDD611.ADA	TFDD722.ADA
TFDD723.ADA	TFDD724.ADA	TFDD725.ADA	TFDD72A.ADA	TFDD72B.ADA	TFDD72C.ADA
TFDD72E.ADA	TFDD72F.ADA	TFDD72G.ADA	TFDD72H.ADA	TFDD72I.ADA	TFDDA01.ADA
TFDDA02.ADA	TFDE000.ADA	TFDE201.ADA	TFDE202.ADA	TFDE203.ADA	TFDE204.ADA
TFDE205.ADA	TFDE206.ADA	TFDE207.ADA	TFDE208.ADA	TFDE209.ADA	TFDE20A.ADA
TFDE211.ADA	TFDE212.ADA	TFDE213.ADA	TFDE214.ADA	TFDE215.ADA	TFDE216.ADA
TFDE220.ADA	TFDE231.ADA	TFDE232.ADA	TFDE233.ADA	TFDE234.ADA	TFDE235.ADA
TFDE236.ADA	TFDE237.ADA	TFDE238.ADA	TFDE239.ADA	TFDE23A.ADA	TFDE241.ADA
TFDE242.ADA	TFDE243.ADA	TFDE244.ADA	TFDE245.ADA	TFDE246.ADA	TFDE250.ADA
TFDE401.ADA	TFDE402.ADA	TFDE403.ADA	TFDE404.ADA	TFDE405.ADA	TFDE406.ADA
TFDE407.ADA	TFDE408.ADA	TFDE409.ADA	TFDE40A.ADA	TFDE411.ADA	TFDE412.ADA
TFDE413.ADA	TFDE414.ADA	TFDE415.ADA	TFDE416.ADA	TFDE420.ADA	TFDE431.ADA
TFDE432.ADA	TFDE433.ADA	TFDE434.ADA	TFDE435.ADA	TFDE436.ADA	TFDE437.ADA
TFDE438.ADA	TFDE439.ADA	TFDE43A.ADA	TFDE441.ADA	TFDE442.ADA	TFDE443.ADA
TFDE444.ADA	TFDE445.ADA	TFDE446.ADA	TFDE450.ADA	TFDF000.ADA	TFDF001.ADA
TFDF002.ADA	TFDF003.ADA	TFDF004.ADA	TFDF005.ADA	TFM9A00.ADA	TFM9A01.ADA
TFM9A02.ADA	TFM9A03.ADA	TFMB000.ADA	TFMB001.ADA	TFMB002.ADA	TFMB003.ADA
TFMB004.ADA	TFMB005.ADA	TFMB006.ADA	TFMB007.ADA	TFMB008.ADA	TFMB009.ADA
TFMB010.ADA	TFMB011.ADA	TFMB012.ADA	TFMB013.ADA	TFMB014.ADA	TFMB015.ADA
TFMB016.ADA	TFMB017.ADA	TFMB018.ADA	TFMB019.ADA	TFMB020.ADA	TFMB021.ADA
TFMB022.ADA	TFMB023.ADA	TFMB024.ADA	TFMB025.ADA	TFMB026.ADA	TFMB027.ADA
TFMB028.ADA	TFMB029.ADA	TFMB030.ADA	TFMB031.ADA	TFMB032.ADA	TFMB033.ADA
TFMB034.ADA	TFMB035.ADA	TFMB036.ADA	TFMB037.ADA	TFMB038.ADA	TFMB039.ADA
TFMB040.ADA	TFMB041.ADA	TFMB042.ADA	TFMB043.ADA	TFMB044.ADA	TFN9300.ADA
TFN9301.ADA	TFN9302.ADA	TFN9303.ADA	TFN9510.ADA	TFN9511.ADA	TFN9610.ADA
TFN9611.ADA	TFN9612.ADA	TFN9613.ADA	TG0000C.ADA	TG0P000.ADA	TGD0000.ADA
TGD0001.ADA	TGD0002.ADA	TGD0003.ADA	TGD0004.ADA	TGD0005.ADA	TGD0006.ADA
TGD0007.ADA	TGD0008.ADA	TGD0009.ADA	TGD000D.ADA	TGD000E.ADA	TGD0019.ADA
TGD0029.ADA	TGD0039.ADA	TGD0049.ADA	TGN000F.ADA	TGN000G.ADA	TGN000H.ADA
TGN000I.ADA	TGN000J.ADA	TGN000K.ADA	TL09000.ADA	TL09100.ADA	TL09101.ADA
TL09111.ADA	TL09121.ADA	TL09131.ADA	TLOP000.ADA	TLD9200.ADA	TLD9202.ADA
TLD9203.ADA	TLD9212.ADA	TLD9213.ADA	TLD9222.ADA	TLD9223.ADA	TLD9232.ADA
TLD9233.ADA	TLDE000.ADA	TLDE2B1.ADA	TLDE2B2.ADA	TLDE2C1.ADA	TLDE2C2.ADA
TLDE2C5.ADA	TLDE2C6.ADA	TLDE4B1.ADA	TLDE4B2.ADA	TLDE4C1.ADA	TLDE4C2.ADA
TLDE4C5.ADA	TLDE4C6.ADA	T000001.ADA	T000002.ADA	T000102.ADA	T000103.ADA
T000104.ADA	T000107.ADA	T000108.ADA	T000109.ADA	T000112.ADA	T000113.ADA
T000114.ADA	T000202.ADA	T000203.ADA	T000204.ADA	T000207.ADA	T000208.ADA
T000209.ADA	T000300.ADA	T000305.ADA	T000310.ADA	T000311.ADA	T000312.ADA
T000313.ADA	T000314.ADA	T000315.ADA	T000316.ADA	T000317.ADA	T000318.ADA
T000319.ADA	T000320.ADA	T000321.ADA	T000322.ADA	T000323.ADA	T000324.ADA
T000325.ADA	T000326.ADA	T000327.ADA	T000328.ADA	T000329.ADA	T000330.ADA
T000331.ADA	T000332.ADA	T000333.ADA	T000334.ADA	T000335.ADA	T000336.ADA
T000337.ADA	T000338.ADA	T000339.ADA	T000340.ADA	T000341.ADA	T000342.ADA
T000343.ADA	T000347.ADA	T000348.ADA	T000349.ADA	T000350.ADA	T000351.ADA
T000352.ADA	T000353.ADA	T000354.ADA	T000355.ADA	T000356.ADA	T000357.ADA
T000358.ADA	T000359.ADA	T000360.ADA	T000361.ADA	T000362.ADA	T000363.ADA
T000364.ADA	T000365.ADA	T000366.ADA	T000367.ADA	T000368.ADA	T000369.ADA
T00036A.ADA	T00036B.ADA	T00036F.ADA	T00036G.ADA	T00036H.ADA	T00036I.ADA
T00036J.ADA	T00036K.ADA	T00036L.ADA	T00036M.ADA	T00036N.ADA	T00036U.ADA
T00036V.ADA	T00036W.ADA	T000374.ADA	T000375.ADA	T000376.ADA	T000377.ADA
T000378.ADA	T000390.ADA	T000391.ADA	T000402.ADA	T000403.ADA	T000404.ADA
T000500.ADA	T000502.ADA	T000503.ADA	T000504.ADA	T000507.ADA	T000508.ADA
T000509.ADA	T000512.ADA	T000513.ADA	T000514.ADA	T000517.ADA	T000518.ADA
T000519.ADA	T000602.ADA	T000603.ADA	T000604.ADA	T000605.ADA	T000606.ADA
T000607.ADA	T000702.ADA	T000703.ADA	T000704.ADA	T000705.ADA	T000706.ADA
T000707.ADA	T000708.ADA	T000709.ADA	T000710.ADA	T000711.ADA	T000713.ADA
T000714.ADA	T000715.ADA	T000716.ADA	T000717.ADA	T000718.ADA	T000719.ADA
T000720.ADA	T000721.ADA	T00P000.ADA	TOD0344.ADA	TOD0345.ADA	TOD0346.ADA
TOD036C.ADA	TOD036D.ADA	TOD036E.ADA	TOD036O.ADA	TOD036P.ADA	TOD036Q.ADA
TOD036R.ADA	TOD036S.ADA	TOD036T.ADA	TOD036X.ADA	TOD036Y.ADA	TOD036Z.ADA
TOD0370.ADA	TOD0371.ADA	TOD0372.ADA	TOD0373.ADA	TOD0380.ADA	TOD0381.ADA
TOD0382.ADA	TOD0383.ADA	TOD0523.ADA	TOD0524.ADA	TOD0525.ADA	TOD0526.ADA
TOD0527.ADA	TOD0528.ADA	TOD0529.ADA	MATHFUN.DEC	ALD9204.DEC	ALD9214.DEC
ALD9224.DEC	ALD9234.DEC	ALDE2C3.DEC	ALDE2C4.DEC	ALDE2C7.DEC	ALDE2C8.DEC
ALDE4C3.DEC	ALDE4C4.DEC	ALDE4C7.DEC	ALDE4C8.DEC	CLD9204.DEC	CLD9214.DEC
CLD9224.DEC	CLD9234.DEC	CLDE2C3.DEC	CLDE2C4.DEC	CLDE2C7.DEC	CLDE2C8.DEC

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

CLDE4C3.DEC	CLDE4C4.DEC	CLDE4C7.DEC	CLDE4C8.DEC	ELD9204.DEC	ELD9214.DEC
ELD9224.DEC	ELD9234.DEC	ELDE2C3.DEC	ELDE2C4.DEC	ELDE2C7.DEC	ELDE2C8.DEC
ELDE4C3.DEC	ELDE4C4.DEC	ELDE4C7.DEC	ELDE4C8.DEC	SLD9204.DEC	SLD9214.DEC
SLD9224.DEC	SLD9234.DEC	SLDE2C3.DEC	SLDE2C4.DEC	SLDE2C7.DEC	SLDE2C8.DEC
SLDE4C3.DEC	SLDE4C4.DEC	SLDE4C7.DEC	SLDE4C8.DEC	TLD9204.DEC	TLD9214.DEC
TLD9224.DEC	TLD9234.DEC	TLDE2C3.DEC	TLDE2C4.DEC	TLDE2C7.DEC	TLDE2C8.DEC
TLDE4C3.DEC	TLDE4C4.DEC	TLDE4C7.DEC	TLDE4C8.DEC	AGD0059.DEC	CGD0059.DEC
EGD0059.DEC	SGD0059.DEC	TGD0059.DEC	AFD9C03.DEC	AFD9C04.DEC	CFD9C03.DEC
CFD9C04.DEC	EFD9C03.DEC	EFD9C04.DEC	SFD9C03.DEC	SFD9C04.DEC	TFD9C03.DEC
TFD9C04.DEC	OURSPC.SPA	OURSYS.SPA	OURSYSR.SPA	OURTYP.SPA	OURTYPR.SPA
OURSPC.TIM	OURSYS.TIM	OURSYSR.TIM	OURTYP.TIM	OURTYPR.TIM	ADASY.SUSE
ADATYP.USE	AFD4128.USE	AFD451U.USE	AFD4803.USE	AFDC107.USE	AFDD608.USE
CFD4128.USE	CFD451U.USE	CFD4803.USE	CFDC107.USE	CFDD608.USE	EFD4128.USE
EFD451U.USE	EFDC107.USE	EFDD608.USE	OURSPC.USE	OURTYP.USE	OURTYPR.USE
SFD4128.USE	SFD451U.USE	SFD4803.USE	SFDC107.USE	SFDD608.USE	TFD4128.USE
TFD451U.USE	TFD4803.USE	TFDC107.USE	TFDD608.USE		
FA0P000.CMN	FF0P000.CMN	FFDE000.CMN	FG0P000.CMN	FGD0000.CMN	FO00002.CMN
FO0P000.CMN	OURFOR.CMN	SYSFOR.CMN	FA00000.FOR	FF03500.FOR	FF03502.FOR
FF03503.FOR	FF03504.FOR	FF03514.FOR	FF03519.FOR	FF03600.FOR	FF03602.FOR
FF03603.FOR	FF03605.FOR	FF03606.FOR	FF03608.FOR	FF03609.FOR	FF03620.FOR
FF04120.FOR	FF04121.FOR	FF04122.FOR	FF04123.FOR	FF04124.FOR	FF04125.FOR
FF04126.FOR	FF04127.FOR	FF04129.FOR	FF04510.FOR	FF04511.FOR	FF04512.FOR
FF04513.FOR	FF04514.FOR	FF04515.FOR	FF04516.FOR	FF04517.FOR	FF04518.FOR
FF04519.FOR	FF0451A.FOR	FF0451B.FOR	FF0451C.FOR	FF0451D.FOR	FF0451E.FOR
FF0451F.FOR	FF0451G.FOR	FF0451H.FOR	FF0451I.FOR	FF0451J.FOR	FF0451V.FOR
FF0451W.FOR	FF0451X.FOR	FF0451Y.FOR	FF0451Z.FOR	FF04520.FOR	FF04521.FOR
FF04522.FOR	FF04529.FOR	FF04530.FOR	FF04531.FOR	FF04532.FOR	FF04533.FOR
FF04534.FOR	FF04535.FOR	FF04536.FOR	FF04537.FOR	FF04538.FOR	FF04539.FOR
FF0453A.FOR	FF04540.FOR	FF04541.FOR	FF04550.FOR	FF04551.FOR	FF04552.FOR
FF04553.FOR	FF04554.FOR	FF04555.FOR	FF04556.FOR	FF04557.FOR	FF04558.FOR
FF04559.FOR	FF04560.FOR	FF04562.FOR	FF04563.FOR	FF04600.FOR	FF04601.FOR
FF04602.FOR	FF04601.FOR	FF05200.FOR	FF05201.FOR	FF05202.FOR	FF05203.FOR
FF05204.FOR	FF05205.FOR	FF05206.FOR	FF05207.FOR	FF05208.FOR	FF05209.FOR
FF0520A.FOR	FF05210.FOR	FF05211.FOR	FF05212.FOR	FF05213.FOR	FF05214.FOR
FF05215.FOR	FF0521H.FOR	FF0521I.FOR	FF0521J.FOR	FF0521K.FOR	FF05300.FOR
FF05301.FOR	FF05302.FOR	FF05303.FOR	FF05304.FOR	FF05305.FOR	FF05306.FOR
FF05307.FOR	FF05308.FOR	FF05400.FOR	FF05401.FOR	FF05402.FOR	FF05403.FOR
FF05404.FOR	FF05405.FOR	FF05406.FOR	FF05407.FOR	FF05408.FOR	FF05501.FOR
FF05502.FOR	FF05503.FOR	FF05504.FOR	FF05505.FOR	FF05506.FOR	FF05507.FOR
FF05508.FOR	FF05509.FOR	FF0550A.FOR	FF0550B.FOR	FF0550C.FOR	FF0550D.FOR
FF0550E.FOR	FF0550F.FOR	FF06001.FOR	FF06009.FOR	FF06010.FOR	FF06011.FOR
FF06013.FOR	FF06014.FOR	FF06015.FOR	FF06016.FOR	FF06017.FOR	FF06018.FOR
FF06019.FOR	FF06022.FOR	FF06023.FOR	FF06024.FOR	FF06025.FOR	FF06026.FOR
FF06027.FOR	FF06028.FOR	FF06029.FOR	FF06030.FOR	FF06031.FOR	FF06032.FOR
FF06033.FOR	FF06043.FOR	FF06044.FOR	FF06045.FOR	FF06046.FOR	FF06047.FOR
FF06048.FOR	FF06049.FOR	FF06050.FOR	FF06051.FOR	FF06052.FOR	FF06053.FOR
FF06060.FOR	FF06061.FOR	FF06062.FOR	FF06063.FOR	FF06064.FOR	FF06065.FOR
FF06066.FOR	FF06067.FOR	FF06068.FOR	FF06069.FOR	FF06070.FOR	FF06071.FOR
FF06072.FOR	FF06073.FOR	FF06074.FOR	FF06075.FOR	FF06076.FOR	FF06077.FOR
FF06078.FOR	FF06079.FOR	FF06100.FOR	FF06101.FOR	FF06108.FOR	FF06109.FOR
FF06110.FOR	FF06111.FOR	FF06112.FOR	FF06113.FOR	FF06114.FOR	FF06115.FOR
FF06116.FOR	FF06117.FOR	FF06118.FOR	FF06119.FOR	FF06122.FOR	FF06132.FOR
FF06133.FOR	FF06134.FOR	FF06135.FOR	FF06136.FOR	FF06137.FOR	FF06138.FOR
FF06139.FOR	FF06140.FOR	FF06142.FOR	FF06143.FOR	FF06144.FOR	FF06145.FOR
FF06146.FOR	FF06147.FOR	FF06148.FOR	FF06149.FOR	FF06150.FOR	FF06152.FOR
FF06153.FOR	FF06154.FOR	FF06155.FOR	FF06156.FOR	FF06157.FOR	FF06158.FOR
FF06159.FOR	FF06160.FOR	FF06162.FOR	FF06163.FOR	FF06164.FOR	FF06165.FOR
FF06166.FOR	FF06167.FOR	FF06168.FOR	FF06169.FOR	FF06170.FOR	FF06172.FOR
FF06173.FOR	FF06174.FOR	FF06175.FOR	FF06176.FOR	FF06177.FOR	FF06178.FOR
FF06179.FOR	FF06180.FOR	FF06182.FOR	FF06183.FOR	FF06184.FOR	FF06185.FOR
FF06186.FOR	FF06187.FOR	FF06188.FOR	FF06189.FOR	FF06190.FOR	FF06802.FOR
FF06803.FOR	FF06806.FOR	FF06807.FOR	FF06816.FOR	FF06817.FOR	FF06818.FOR
FF06819.FOR	FF06820.FOR	FF06821.FOR	FF0D720.FOR	FF0D721.FOR	FF0D727.FOR
FF0D728.FOR	FFD451K.FOR	FFD451L.FOR	FFD451M.FOR	FFD451N.FOR	FFD451O.FOR
FFD451P.FOR	FFD451Q.FOR	FFD451R.FOR	FFD451S.FOR	FFD4523.FOR	FFD4524.FOR
FFD4525.FOR	FFD4526.FOR	FFD455D.FOR	FFD455E.FOR	FFD455F.FOR	FFD455G.FOR
FFD455I.FOR	FFD455J.FOR	FFD455K.FOR	FFD455L.FOR	FFD4567.FOR	FFD4568.FOR

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

```
FFD4569.FOR FFD4603.FOR FFD4604.FOR FFD4605.FOR FFD4606.FOR FFD4607.FOR
FFD4608.FOR FFD4609.FOR FFD460A.FOR FFD460B.FOR FFD520C.FOR FFD520D.FOR
FFD520E.FOR FFD520F.FOR FFD520G.FOR FFD520H.FOR FFD520I.FOR FFD520J.FOR
FFD521E.FOR FFD521F.FOR FFD521G.FOR FFD521H.FOR FFD521I.FOR FFD521J.FOR
FFDE201.FOR FFDE202.FOR FFDE203.FOR FFDE204.FOR FFDE205.FOR FFDE206.FOR
FFDE207.FOR FFDE208.FOR FFDE209.FOR FFDE20A.FOR FFDE211.FOR FFDE212.FOR
FFDE213.FOR FFDE214.FOR FFDE215.FOR FFDE216.FOR FFDE220.FOR FFDE231.FOR
FFDE232.FOR FFDE233.FOR FFDE234.FOR FFDE235.FOR FFDE236.FOR FFDE237.FOR
FFDE238.FOR FFDE239.FOR FFDE23A.FOR FFDE241.FOR FFDE242.FOR FFDE243.FOR
FFDE244.FOR FFDE245.FOR FFDE246.FOR FFDE250.FOR FFDE401.FOR FFDE402.FOR
FFDE403.FOR FFDE404.FOR FFDE405.FOR FFDE406.FOR FFDE407.FOR FFDE408.FOR
FFDE409.FOR FFDE40A.FOR FFDE411.FOR FFDE412.FOR FFDE413.FOR FFDE414.FOR
FFDE415.FOR FFDE416.FOR FFDE420.FOR FFDE431.FOR FFDE432.FOR FFDE433.FOR
FFDE434.FOR FFDE435.FOR FFDE436.FOR FFDE437.FOR FFDE438.FOR FFDE439.FOR
FFDE43A.FOR FFDE441.FOR FFDE442.FOR FFDE443.FOR FFDE444.FOR FFDE445.FOR
FFDE446.FOR FFDE450.FOR FFDF000.FOR FFDF001.FOR FFDF002.FOR FFDF003.FOR
FFDF004.FOR FFDF005.FOR FFIRST.FOR FGDD001.FOR FGDD00D.FOR FGDD00E.FOR
FO00102.FOR FO00103.FOR FO00104.FOR FO00202.FOR FO00203.FOR FO00204.FOR
FO00300.FOR FO00305.FOR FO00310.FOR FO00311.FOR FO00312.FOR FO00313.FOR
FO00314.FOR FO00315.FOR FO00316.FOR FO00317.FOR FO00318.FOR FO00319.FOR
FO00320.FOR FO00321.FOR FO00322.FOR FO00323.FOR FO00324.FOR FO00325.FOR
FO00326.FOR FO00327.FOR FO00328.FOR FO00329.FOR FO00330.FOR FO00331.FOR
FO00332.FOR FO00333.FOR FO00334.FOR FO00335.FOR FO00336.FOR FO00337.FOR
FO00338.FOR FO00339.FOR FO00340.FOR FO00341.FOR FO00342.FOR FO00343.FOR
FO00347.FOR FO00348.FOR FO00349.FOR FO00350.FOR FO00351.FOR FO00352.FOR
FO00353.FOR FO00354.FOR FO00355.FOR FO00356.FOR FO00357.FOR FO00358.FOR
FO00359.FOR FO00360.FOR FO00361.FOR FO00362.FOR FO00363.FOR FO00364.FOR
FO00402.FOR FO00403.FOR FO00404.FOR FO00508.FOR FO00509.FOR FO00510.FOR
FO00514.FOR FO00517.FOR FO00518.FOR FO00519.FOR FO00602.FOR FO00603.FOR
FO00604.FOR FO00605.FOR FO00606.FOR FO00607.FOR FO00702.FOR FO00703.FOR
FO00704.FOR FO00705.FOR FO00711.FOR FO00713.FOR FO00714.FOR FO00715.FOR
FO00717.FOR FO00718.FOR FO00719.FOR FO00720.FOR FO00721.FOR FOD0344.FOR
FOD0345.FOR FOD0346.FOR FOD036X.FOR FOD036Y.FOR FOD036Z.FOR FORDMP.FOR
FORSYS.FOR SYSBLK.FOR FFD451U.USE FFD456Y.USE FORTYP.USE TYFPOR.USE
JAOP000.CPL JFOP000.CPL JFJ3500.CPL JFJ3501.CPL JFJ3502.CPL JFJ3503.CPL
JFJ3504.CPL JFJ3513.CPL JFJ3514.CPL JFJ3600.CPL JFJ3601.CPL JFJ3602.CPL
JFJ3603.CPL JFJ3604.CPL JFJ3605.CPL JFJ3606.CPL JFJ3607.CPL JFJ3608.CPL
JFJ3609.CPL JFJ3610.CPL JFJ3700.CPL JFJ3701.CPL JFJ3702.CPL JFJ3703.CPL
JFJ3800.CPL JFJ3801.CPL JFJ3802.CPL JFJ3803.CPL JFJ3804.CPL JFJ4518.CPL
JFJ4519.CPL JFJ451A.CPL JFJ451B.CPL JFJ451C.CPL JFJ451D.CPL JFJ451E.CPL
JFJ451F.CPL JFJ451G.CPL JFJ451H.CPL JFJ451I.CPL JFJ451J.CPL JFJ451K.CPL
JFJ451L.CPL JFJ451M.CPL JFJ451N.CPL JFJ451O.CPL JFJ451P.CPL JFJ451Q.CPL
JFJ451R.CPL JFJ451S.CPL JFJ451T.CPL JFJ451U.CPL JFJ451V.CPL JFJ451W.CPL
JFJ451X.CPL JFJ451Y.CPL JFJ451Z.CPL JFJ5300.CPL JFJ5301.CPL JFJ5302.CPL
JFJ5303.CPL JFJ5304.CPL JFJ5305.CPL JFJ5306.CPL JFJ5307.CPL JFJ5308.CPL
JFJ5400.CPL JFJ5401.CPL JFJ5402.CPL JFJ5403.CPL JFJ5404.CPL JFJ5405.CPL
JFJ5406.CPL JFJ5407.CPL JFJ5408.CPL JFJ5409.CPL JFJ5410.CPL JFJ5411.CPL
JFJ5412.CPL JFJ5413.CPL JFJ5414.CPL JFJ5415.CPL JFJ5416.CPL JFJ5417.CPL
JFJ5418.CPL JFJ5419.CPL JFJ5420.CPL JFJ5421.CPL JFJ5422.CPL JFJ5423.CPL
JFJ5424.CPL JFJ5425.CPL JFJ5426.CPL JFJ5427.CPL JFJ5428.CPL JFJ5429.CPL
JFJ5430.CPL JFJ5431.CPL JFJ5432.CPL JFJ5433.CPL JFJ5434.CPL JFJ5435.CPL
JFJ5436.CPL JFJ5437.CPL JFJ5438.CPL JFJ5439.CPL JFJ5440.CPL JFJ5441.CPL
JFJ5442.CPL JFJ5443.CPL JFJ5444.CPL JFJ5445.CPL JFJ5446.CPL JFJ5447.CPL
JFJ5448.CPL JFJ5449.CPL JFJ5450.CPL JFJ5451.CPL JFJ5452.CPL JFJ5453.CPL
JFJ5454.CPL JFJ5455.CPL JFJ5456.CPL JFJ5457.CPL JFJ5458.CPL JFJ5459.CPL
JFJ5460.CPL JFJ5461.CPL JFJ5462.CPL JFJ5463.CPL JFJ5464.CPL JFJ5465.CPL
JFJ5466.CPL JFJ5467.CPL JFJ5468.CPL JFJ5469.CPL JFJ5470.CPL JFJ5471.CPL
JFJ5472.CPL JFJ5473.CPL JFJ5474.CPL JFJ5475.CPL JFJ5476.CPL JFJ5477.CPL
JFJ5478.CPL JFJ5479.CPL JFJ5480.CPL JFJ5481.CPL JFJ5482.CPL JFJ5483.CPL
JFJ5484.CPL JFJ5485.CPL JFJ5486.CPL JFJ5487.CPL JFJ5488.CPL JFJ5489.CPL
JFJ5490.CPL JFJ5491.CPL JFJ5492.CPL JFJ5493.CPL JFJ5494.CPL JFJ5495.CPL
```

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

JFJ6196.CPL	JFJ6197.CPL	JFJ6198.CPL	JFJ619A.CPL	JFJ619B.CPL	JFJ619C.CPL
JFJ619D.CPL	JFJ619E.CPL	JFJ619F.CPL	JFJ619G.CPL	JFJ619H.CPL	JFJ619J.CPL
JFJ619K.CPL	JFJ619L.CPL	JFJ619M.CPL	JFJ619N.CPL	JFJ619O.CPL	JFJ619P.CPL
JFJ619Q.CPL	JFJ6213.CPL	JFJ6214.CPL	JFJ6215.CPL	JFJ6216.CPL	JFJ6217.CPL
JFJ6218.CPL	JFJ6219.CPL	JFJ6223.CPL	JFJ6224.CPL	JFJ6225.CPL	JFJ6226.CPL
JFJ6227.CPL	JFJ6228.CPL	JFJ6229.CPL	JFJ6230.CPL	JFJ6231.CPL	JFJ6232.CPL
JFJ6243.CPL	JFJ6244.CPL	JFJ6245.CPL	JFJ6246.CPL	JFJ6247.CPL	JFJ6248.CPL
JFJ6249.CPL	JFJ6250.CPL	JFJ6251.CPL	JFJ6252.CPL	JFJ6802.CPL	JFJ6806.CPL
JFJ6808.CPL	JFJ680A.CPL	JFJ680C.CPL	JFJ680E.CPL	JFJ680G.CPL	JFJ680I.CPL
JFJ6810.CPL	JFJ6814.CPL	JFJ6816.CPL	JFJ6818.CPL	JFJ6820.CPL	JFJ6822.CPL
JFJ6824.CPL	JGD0P000.CPL	JGD0000.CPL	JGD0004.CPL	JGD0005.CPL	JGD0006.CPL
JGJ000D.CPL	JGN000F.CPL	JGN000G.CPL	JGN000H.CPL	J000002.CPL	J00P000.CPL
JOJ0102.CPL	JOJ0103.CPL	JOJ0202.CPL	JOJ0203.CPL	JOJ0311.CPL	JOJ0312.CPL
JOJ0314.CPL	JOJ0315.CPL	JOJ0323.CPL	JOJ0324.CPL	JOJ0326.CPL	JOJ0327.CPL
JOJ0329.CPL	JOJ0330.CPL	JOJ0332.CPL	JOJ0333.CPL	JOJ0335.CPL	JOJ0336.CPL
JOJ0338.CPL	JOJ0339.CPL	JOJ0341.CPL	JOJ0342.CPL	JOJ0344.CPL	JOJ0345.CPL
JOJ0347.CPL	JOJ0348.CPL	JOJ0349.CPL	JOJ0351.CPL	JOJ0352.CPL	JOJ0363.CPL
JOJ0364.CPL	JOJ036I.CPL	JOJ036J.CPL	JOJ036L.CPL	JOJ036M.CPL	JOJ036O.CPL
JOJ036P.CPL	JOJ036R.CPL	JOJ036S.CPL	JOJ036U.CPL	JOJ036V.CPL	JOJ036X.CPL
JOJ036Y.CPL	JOJ0390.CPL	JOJ0402.CPL	JOJ0403.CPL	JOJ0502.CPL	JOJ0503.CPL
JOJ0507.CPL	JOJ0508.CPL	JOJ0512.CPL	JOJ0513.CPL	JOJ0517.CPL	JOJ0518.CPL
JOJ0523.CPL	JOJ0524.CPL	JOJ0602.CPL	JOJ0603.CPL	JOJ0702.CPL	JOJ0703.CPL
JOJ0706.CPL	JOJ0707.CPL	JOJ0709.CPL	JOJ0710.CPL	JOJ0719.CPL	JOJ0720.CPL
JOVDMP.CPL	JOVSPC.CPL	OURJOV.CPL	DMPJOV.JOV	JA00000.JOV	JF03500.JOV
JF03501.JOV	JF03502.JOV	JF03503.JOV	JF03504.JOV	JF03513.JOV	JF03514.JOV
JF03519.JOV	JF03550.JOV	JF03600.JOV	JF03601.JOV	JF03602.JOV	JF03603.JOV
JF03604.JOV	JF03605.JOV	JF03606.JOV	JF03607.JOV	JF03608.JOV	JF03609.JOV
JF03610.JOV	JF03620.JOV	JF03700.JOV	JF03701.JOV	JF03702.JOV	JF03703.JOV
JF03704.JOV	JF03800.JOV	JF03801.JOV	JF03802.JOV	JF03803.JOV	JF03804.JOV
JF03805.JOV	JF04120.JOV	JF04121.JOV	JF04122.JOV	JF04123.JOV	JF04124.JOV
JF04125.JOV	JF04126.JOV	JF04127.JOV	JF04129.JOV	JF0412B.JOV	JF04130.JOV
JF04131.JOV	JF04132.JOV	JF04133.JOV	JF0413D.JOV	JF0413H.JOV	JF0413I.JOV
JF0413J.JOV	JF0413K.JOV	JF04310.JOV	JF04311.JOV	JF04312.JOV	JF04510.JOV
JF04511.JOV	JF04512.JOV	JF04513.JOV	JF04514.JOV	JF04515.JOV	JF04516.JOV
JF04517.JOV	JF04518.JOV	JF04519.JOV	JF0451A.JOV	JF0451B.JOV	JF0451C.JOV
JF0451D.JOV	JF0451E.JOV	JF0451F.JOV	JF0451G.JOV	JF0451J.JOV	JF0451V.JOV
JF0451H.JOV	JF0451X.JOV	JF0451Y.JOV	JF0451Z.JOV	JF04520.JOV	JF04521.JOV
JF04522.JOV	JF04527.JOV	JF04529.JOV	JF04530.JOV	JF04531.JOV	JF04532.JOV
JF04533.JOV	JF04534.JOV	JF04535.JOV	JF04536.JOV	JF04537.JOV	JF04538.JOV
JF04539.JOV	JF0453A.JOV	JF0453B.JOV	JF0453C.JOV	JF04540.JOV	JF04541.JOV
JF04550.JOV	JF04551.JOV	JF04552.JOV	JF04553.JOV	JF04554.JOV	JF04555.JOV
JF04556.JOV	JF04557.JOV	JF04558.JOV	JF04559.JOV	JF0455N.JOV	JF0455O.JOV
JF04560.JOV	JF04562.JOV	JF04563.JOV	JF04600.JOV	JF04601.JOV	JF04602.JOV
JF0460E.JOV	JF0460F.JOV	JF0460G.JOV	JF0460H.JOV	JF0460I.JOV	JF0460J.JOV
JF0460M.JOV	JF05200.JOV	JF05201.JOV	JF05202.JOV	JF05203.JOV	JF05204.JOV
JF05205.JOV	JF05206.JOV	JF05207.JOV	JF05208.JOV	JF05209.JOV	JF0520A.JOV
JF0520B.JOV	JF0520M.JOV	JF05200.JOV	JF0520R.JOV	JF0520S.JOV	JF05210.JOV
JF05211.JOV	JF05212.JOV	JF05213.JOV	JF05214.JOV	JF05215.JOV	JF0521H.JOV
JF0521I.JOV	JF0521J.JOV	JF0521K.JOV	JF0521L.JOV	JF0521N.JOV	JF05300.JOV
JF05301.JOV	JF05302.JOV	JF05303.JOV	JF05304.JOV	JF05305.JOV	JF05306.JOV
JF05307.JOV	JF05308.JOV	JF05400.JOV	JF05401.JOV	JF05402.JOV	JF05403.JOV
JF05404.JOV	JF05405.JOV	JF05406.JOV	JF05407.JOV	JF05408.JOV	JF05501.JOV
JF05502.JOV	JF05503.JOV	JF05504.JOV	JF05505.JOV	JF05506.JOV	JF05507.JOV
JF05508.JOV	JF05509.JOV	JF0550A.JOV	JF0550B.JOV	JF0550C.JOV	JF0550D.JOV
JF0550E.JOV	JF0550F.JOV	JF06001.JOV	JF06009.JOV	JF06010.JOV	JF06011.JOV
JF06013.JOV	JF06014.JOV	JF06015.JOV	JF06016.JOV	JF06017.JOV	JF06018.JOV
JF06019.JOV	JF06022.JOV	JF06023.JOV	JF06024.JOV	JF06025.JOV	JF06026.JOV
JF06027.JOV	JF06028.JOV	JF06029.JOV	JF06030.JOV	JF06031.JOV	JF06032.JOV
JF06033.JOV	JF06043.JOV	JF06044.JOV	JF06045.JOV	JF06046.JOV	JF06047.JOV
JF06048.JOV	JF06049.JOV	JF06050.JOV	JF06051.JOV	JF06052.JOV	JF06053.JOV
JF06060.JOV	JF06061.JOV	JF06062.JOV	JF06063.JOV	JF06064.JOV	JF06065.JOV
JF06066.JOV	JF06067.JOV	JF06068.JOV	JF06069.JOV	JF06070.JOV	JF06071.JOV
JF06072.JOV	JF06073.JOV	JF06074.JOV	JF06075.JOV	JF06076.JOV	JF06077.JOV
JF06078.JOV	JF06079.JOV	JF06100.JOV	JF06101.JOV	JF06108.JOV	JF06109.JOV
JF06110.JOV	JF06111.JOV	JF06112.JOV	JF06113.JOV	JF06114.JOV	JF06115.JOV
JF06116.JOV	JF06117.JOV	JF06118.JOV	JF06119.JOV	JF06122.JOV	JF06132.JOV
JF06133.JOV	JF06134.JOV	JF06135.JOV	JF06136.JOV	JF06137.JOV	JF06138.JOV

Table A-1 ACPS ANSI Tape Contents Volume ACPS (continued)

JF06139.JOV	JF06140.JOV	JF06142.JOV	JF06143.JOV	JF06144.JOV	JF06145.JOV
JF06146.JOV	JF06147.JOV	JF06148.JOV	JF06149.JOV	JF06150.JOV	JF06152.JOV
JF06153.JOV	JF06154.JOV	JF06155.JOV	JF06156.JOV	JF06157.JOV	JF06158.JOV
JF06159.JOV	JF06160.JOV	JF06162.JOV	JF06163.JOV	JF06164.JOV	JF06165.JOV
JF06166.JOV	JF06167.JOV	JF06168.JOV	JF06169.JOV	JF06170.JOV	JF06172.JOV
JF06173.JOV	JF06174.JOV	JF06175.JOV	JF06176.JOV	JF06177.JOV	JF06178.JOV
JF06179.JOV	JF06180.JOV	JF06182.JOV	JF06183.JOV	JF06184.JOV	JF06185.JOV
JF06186.JOV	JF06187.JOV	JF06188.JOV	JF06189.JOV	JF06190.JOV	JF06191.JOV
JF06192.JOV	JF06193.JOV	JF06194.JOV	JF06195.JOV	JF06196.JOV	JF06197.JOV
JF06198.JOV	JF06199.JOV	JF0619A.JOV	JF0619B.JOV	JF0619C.JOV	JF0619D.JOV
JF0619E.JOV	JF0619F.JOV	JF0619G.JOV	JF0619H.JOV	JF0619I.JOV	JF0619J.JOV
JF0619K.JOV	JF0619L.JOV	JF0619M.JOV	JF0619N.JOV	JF0619O.JOV	JF0619P.JOV
JF0619Q.JOV	JF0619R.JOV	JF06802.JOV	JF06803.JOV	JF06806.JOV	JF06807.JOV
JF06808.JOV	JF06809.JOV	JF0680A.JOV	JF0680B.JOV	JF0680C.JOV	JF0680D.JOV
JF0680E.JOV	JF0680F.JOV	JF06810.JOV	JF06811.JOV	JF06814.JOV	JF06815.JOV
JF06816.JOV	JF06817.JOV	JF06818.JOV	JF06819.JOV	JF06820.JOV	JF06821.JOV
JF06822.JOV	JF06823.JOV	JF06824.JOV	JF06825.JOV	JF0D720.JOV	JF0D721.JOV
JF0D727.JOV	JF0D728.JOV	JF0D729.JOV	JF0D72D.JOV	JFD3551.JOV	JFD4128.JOV
JFD412A.JOV	JFD412C.JOV	JFD412D.JOV	JFD4135.JOV	JFD4136.JOV	JFD4137.JOV
JFD4138.JOV	JFD4139.JOV	JFD413A.JOV	JFD413B.JOV	JFD413C.JOV	JFD413E.JOV
JFD413G.JOV	JFD451K.JOV	JFD451L.JOV	JFD451M.JOV	JFD451P.JOV	JFD451Q.JOV
JFD4523.JOV	JFD4524.JOV	JFD452A.JOV	JFD455D.JOV	JFD455E.JOV	JFD455I.JOV
JFD455J.JOV	JFD4566.JOV	JFD4567.JOV	JFD4603.JOV	JFD4604.JOV	JFD4605.JOV
JFD4606.JOV	JFD4607.JOV	JFD460K.JOV	JFD460L.JOV	JFD460N.JOV	JFD460O.JOV
JFD520C.JOV	JFD520D.JOV	JFD520E.JOV	JFD520F.JOV	JFD520N.JOV	JFD520P.JOV
JFD520Q.JOV	JFD5218.JOV	JFD5219.JOV	JFD521A.JOV	JFD521B.JOV	JFD521K.JOV
JFD521M.JOV	JFD521O.JOV	JFD521P.JOV	JFD6201.JOV	JFD6209.JOV	JFD6210.JOV
JFD6211.JOV	JFD6213.JOV	JFD6214.JOV	JFD6215.JOV	JFD6216.JOV	JFD6217.JOV
JFD6218.JOV	JFD6219.JOV	JFD6222.JOV	JFD6223.JOV	JFD6224.JOV	JFD6225.JOV
JFD6226.JOV	JFD6227.JOV	JFD6228.JOV	JFD6229.JOV	JFD6230.JOV	JFD6231.JOV
JFD6232.JOV	JFD6233.JOV	JFD6243.JOV	JFD6244.JOV	JFD6245.JOV	JFD6246.JOV
JFD6247.JOV	JFD6248.JOV	JFD6249.JOV	JFD6250.JOV	JFD6251.JOV	JFD6252.JOV
JFD6253.JOV	JFD680G.JOV	JFD680H.JOV	JFD680I.JOV	JFD680J.JOV	JFDD600.JOV
JFDD601.JOV	JFDD602.JOV	JFDD603.JOV	JFDD604.JOV	JFDD605.JOV	JFDD606.JOV
JFDD607.JOV	JFDD608.JOV	JFDD609.JOV	JFDD610.JOV	JFDD611.JOV	JFDD722.JOV
JFDD723.JOV	JFDD72A.JOV	JFDD72B.JOV	JFDD72C.JOV	JFDD72E.JOV	JFDD72F.JOV
JFDD72G.JOV	JFDD72H.JOV	JFDD72I.JOV	JFDDA01.JOV	JFDDA02.JOV	JFDF000.JOV
JFDF001.JOV	JFDF002.JOV	JFDF003.JOV	JFDF004.JOV	JFDF005.JOV	JFIRST.JOV
JGD0000C.JOV	JGD0001.JOV	JGD0007.JOV	JGD0008.JOV	JGD0009.JOV	JGD000D.JOV
JGD000E.JOV	JGN0001.JOV	JGN000J.JOV	JGN000K.JOV	J000102.JOV	J000103.JOV
J000104.JOV	J000202.JOV	J000203.JOV	J000204.JOV	J000300.JOV	J000305.JOV
J000310.JOV	J000311.JOV	J000312.JOV	J000313.JOV	J000314.JOV	J000315.JOV
J000316.JOV	J000323.JOV	J000324.JOV	J000325.JOV	J000326.JOV	J000327.JOV
J000328.JOV	J000329.JOV	J000330.JOV	J000331.JOV	J000332.JOV	J000333.JOV
J000334.JOV	J000335.JOV	J000336.JOV	J000337.JOV	J000338.JOV	J000339.JOV
J000340.JOV	J000341.JOV	J000342.JOV	J000343.JOV	J000347.JOV	J000348.JOV
J000349.JOV	J000350.JOV	J000351.JOV	J000352.JOV	J000353.JOV	J000363.JOV
J000364.JOV	J000365.JOV	J00036I.JOV	J00036J.JOV	J00036K.JOV	J00036L.JOV
J00036M.JOV	J00036N.JOV	J00036U.JOV	J00036V.JOV	J00036W.JOV	J000390.JOV
J000391.JOV	J000402.JOV	J000403.JOV	J000404.JOV	J000500.JOV	J000502.JOV
J000503.JOV	J000504.JOV	J000507.JOV	J000508.JOV	J000509.JOV	J000512.JOV
J000513.JOV	J000514.JOV	J000517.JOV	J000518.JOV	J000519.JOV	J000602.JOV
J000603.JOV	J000604.JOV	J000605.JOV	J000606.JOV	J000607.JOV	J000702.JOV
J000703.JOV	J000704.JOV	J000705.JOV	J000706.JOV	J000707.JOV	J000708.JOV
J000709.JOV	J000710.JOV	J000711.JOV	J000713.JOV	J000714.JOV	J000715.JOV
J000716.JOV	J000717.JOV	J000718.JOV	J000719.JOV	J000720.JOV	J000721.JOV
J0D0344.JOV	J0D0345.JOV	J0D0346.JOV	J0D0360.JOV	J0D036P.JOV	J0D036Q.JOV
J0D036R.JOV	J0D036S.JOV	J0D036T.JOV	J0D036X.JOV	J0D036Y.JOV	J0D036Z.JOV
J0D0523.JOV	J0D0524.JOV	J0D0525.JOV	J0VSYS.JOV	SPCJOV.JOV	JF03701.USE
JF04311.USE	JF0451H.USE	JF0451I.USE	JF0451J.USE	JF0619A.USE	JF06822.USE
JFD451N.USE	JFD451O.USE	JFD451R.USE	JFD451S.USE	JFD451U.USE	JFD4525.USE
JFD4526.USE	JFD4528.USE	JFD455F.USE	JFD455G.USE	JFD455K.USE	JFD455L.USE
JFD4568.USE	JFD4569.USE	JFD4608.USE	JFD4609.USE	JFD460A.USE	JFD460B.USE
JFD460C.USE	JFD460D.USE	JFD520G.USE	JFD520H.USE	JFD520I.USE	JFD520J.USE
JFD521C.USE	JFD521D.USE	JFD521E.USE	JFD521F.USE	JFD6260.USE	JFD6261.USE
JFD6262.USE	JFD6263.USE	JFD6264.USE	JFD6265.USE	JFD6266.USE	JFD6267.USE
JFD6268.USE	JFD6269.USE	JFD6270.USE	JFD6271.USE	JFD6272.USE	JFD6273.USE

Table A-1 ACPS ANSI Tape Contents Volume ACPS (concluded)

JFD6274.USE	JFD6275.USE	JFD6276.USE	JFD6277.USE	JFD6278.USE	JFD6279.USE
JFDD602.USE	JFDD603.USE	JFDD610.USE	JFDD611.USE	JFDD724.USE	JFDD725.USE
JFJ451H.USE	JFJ451I.USE	JFJ451N.USE	JFJ451O.USE	JFJ451R.USE	JFJ451S.USE
JFJ6260.USE	JFJ6261.USE	JFJ6262.USE	JFJ6263.USE	JFJ6264.USE	JFJ6265.USE
JFJ6266.USE	JFJ6267.USE	JFJ6268.USE	JFJ6270.USE	JFJ6271.USE	JFJ6272.USE
JFJ6273.USE	JFJ6274.USE	JFJ6275.USE	JFJ6276.USE	JFJ6277.USE	JFJ6278.USE
JGD0008.USE	JGN000J.USE	J000317.USE	J000318.USE	J000319.USE	J000320.USE
J000321.USE	J000322.USE	J0J0317.USE	J0J0318.USE	J0J0320.USE	J0J0321.USE
JOVTYP.USE	TYPJOV.USE	JGF0001.FOR	JGF000E.FOR	SYSJOV.FOR	TMPDMP.FOR
JOVMATH.DAT					
CCOMPA.CMN	CCOMPV.CMN	CEXECA.CMN	CEXECV.CMN	CCOMP.FOR	CEXEC.FOR

Table A-2. ACPS ANSI Tape Contents Volume ACPS02

CF06178.ADA	CF06179.ADA	CF06180.ADA	CF06182.ADA	CF06183.ADA	CF06184.ADA
CF06185.ADA	CF06186.ADA	CF06187.ADA	CF06188.ADA	CF06189.ADA	CF06190.ADA
CF06191.ADA	CF06192.ADA	CF06193.ADA	CF06194.ADA	CF06195.ADA	CF06196.ADA
CF06197.ADA	CF06198.ADA	CF06199.ADA	CF0619A.ADA	CF0619B.ADA	CF0619C.ADA
CF0619D.ADA	CF0619E.ADA	CF0619F.ADA	CF0619G.ADA	CF0619H.ADA	CF0619I.ADA
CF0619J.ADA	CF0619K.ADA	CF0619L.ADA	CF0619M.ADA	CF0619N.ADA	CF0619O.ADA
CF0619P.ADA	CF0619Q.ADA	CF0619R.ADA	CF06422.ADA	CF06423.ADA	CF06424.ADA
CF06425.ADA	CF06426.ADA	CF06427.ADA	CF06428.ADA	CF06429.ADA	CF0642A.ADA
CF0642B.ADA	CF06802.ADA	CF06803.ADA	CF06806.ADA	CF06807.ADA	CF06808.ADA
CF06809.ADA	CF0680A.ADA	CF0680B.ADA	CF0680C.ADA	CF0680D.ADA	CF0680E.ADA
CF0680F.ADA	CF06810.ADA	CF06811.ADA	CF06814.ADA	CF06815.ADA	CF06816.ADA
CF06817.ADA	CF06818.ADA	CF06819.ADA	CF06820.ADA	CF06821.ADA	CF06822.ADA
CF06823.ADA	CF06824.ADA	CF06825.ADA	CF09500.ADA	CF09501.ADA	CF09502.ADA
CF09503.ADA	CF09504.ADA	CF09505.ADA	CF09506.ADA	CF09507.ADA	CF09508.ADA
CF09509.ADA	CF09600.ADA	CF09601.ADA	CF09602.ADA	CF09603.ADA	CF09604.ADA
CF09605.ADA	CF09606.ADA	CF09607.ADA	CF09710.ADA	CF09711.ADA	CF09712.ADA
CF09713.ADA	CF09720.ADA	CF09721.ADA	CF09731.ADA	CF09900.ADA	CF09901.ADA
CF09902.ADA	CF09903.ADA	CF09801.ADA	CF0C100.ADA	CF0C101.ADA	CF0C102.ADA
CF0C103.ADA	CF0C300.ADA	CF0C301.ADA	CF0C302.ADA	CF0C303.ADA	CF0C304.ADA
CF0C305.ADA	CF0C306.ADA	CF0C307.ADA	CF0P000.ADA	CFD3551.ADA	CFD412A.ADA
CF04132.ADA	CFD412D.ADA	CFD4135.ADA	CFD4136.ADA	CFD4137.ADA	CFD4138.ADA
CFD4139.ADA	CFD413A.ADA	CFD413B.ADA	CFD413C.ADA	CFD413E.ADA	CFD413G.ADA
CFD451K.ADA	CFD451L.ADA	CFD451M.ADA	CFD451N.ADA	CFD451O.ADA	CFD451P.ADA
CFD451Q.ADA	CFD451R.ADA	CFD451S.ADA	CFD4523.ADA	CFD4524.ADA	CFD4525.ADA
CFD4526.ADA	CFD4528.ADA	CFD452A.ADA	CFD455D.ADA	CFD455E.ADA	CFD455F.ADA
CFD455G.ADA	CFD455I.ADA	CFD455J.ADA	CFD455K.ADA	CFD455L.ADA	CFD4566.ADA
CFD4567.ADA	CFD4568.ADA	CFD4569.ADA	CFD4603.ADA	CFD4604.ADA	CFD4605.ADA
CFD4606.ADA	CFD4607.ADA	CFD4608.ADA	CFD4609.ADA	CFD460A.ADA	CFD460B.ADA
CFD460K.ADA	CFD460L.ADA	CFD460N.ADA	CFD460O.ADA	CFD4800.ADA	CFD4801.ADA
CFD4802.ADA	CFD4803.ADA	CFD4804.ADA	CFD520C.ADA	CFD520D.ADA	CFD520E.ADA
CFD520F.ADA	CFD520G.ADA	CFD520H.ADA	CFD520I.ADA	CFD520J.ADA	CFD520N.ADA
CFD520P.ADA	CFD520Q.ADA	CFD5218.ADA	CFD5219.ADA	CFD521A.ADA	CFD521B.ADA
CFD521C.ADA	CFD521D.ADA	CFD521E.ADA	CFD521F.ADA	CFD521K.ADA	CFD521M.ADA
CFD521O.ADA	CFD521P.ADA	CFD6201.ADA	CFD6209.ADA	CFD6210.ADA	CFD6211.ADA
CFD6213.ADA	CFD6214.ADA	CFD6215.ADA	CFD6216.ADA	CFD6217.ADA	CFD6218.ADA
CFD6219.ADA	CFD6222.ADA	CFD6223.ADA	CFD6224.ADA	CFD6225.ADA	CFD6226.ADA
CFD6227.ADA	CFD6228.ADA	CFD6229.ADA	CFD6230.ADA	CFD6231.ADA	CFD6232.ADA
CFD6233.ADA	CFD6243.ADA	CFD6244.ADA	CFD6245.ADA	CFD6246.ADA	CFD6247.ADA
CFD6248.ADA	CFD6249.ADA	CFD6250.ADA	CFD6251.ADA	CFD6252.ADA	CFD6253.ADA
CFD6260.ADA	CFD6261.ADA	CFD6262.ADA	CFD6263.ADA	CFD6264.ADA	CFD6265.ADA
CFD6266.ADA	CFD6267.ADA	CFD6268.ADA	CFD6269.ADA	CFD6270.ADA	CFD6271.ADA
CFD6272.ADA	CFD6273.ADA	CFD6274.ADA	CFD6275.ADA	CFD6276.ADA	CFD6277.ADA
CFD6278.ADA	CFD6279.ADA	CFD6300.ADA	CFD6301.ADA	CFD6308.ADA	CFD6309.ADA
CFD6310.ADA	CFD6311.ADA	CFD6312.ADA	CFD6313.ADA	CFD6314.ADA	CFD6315.ADA
CFD6316.ADA	CFD6317.ADA	CFD6318.ADA	CFD6319.ADA	CFD6322.ADA	CFD6332.ADA
CFD6333.ADA	CFD6334.ADA	CFD6335.ADA	CFD6336.ADA	CFD6337.ADA	CFD6338.ADA
CFD6339.ADA	CFD6340.ADA	CFD6342.ADA	CFD6343.ADA	CFD6344.ADA	CFD6345.ADA
CFD6346.ADA	CFD6347.ADA	CFD6348.ADA	CFD6349.ADA	CFD6350.ADA	CFD6352.ADA
CFD6353.ADA	CFD6354.ADA	CFD6355.ADA	CFD6356.ADA	CFD6357.ADA	CFD6358.ADA
CFD6359.ADA	CFD6				

Table A-2. ACPS ANSI Tape Contents Volume ACPS02 (Continued)

CFDE43A.ADA	CFDE441.ADA	CFDE442.ADA	CFDE443.ADA	CFDE444.ADA	CFDE445.ADA
CFDE446.ADA	CFDE450.ADA	CFDF000.ADA	CFDF001.ADA	CFDF002.ADA	CFDF003.ADA
CFDF004.ADA	CFDF005.ADA	CFM9A00.ADA	CFM9A01.ADA	CFM9A02.ADA	CFM9A03.ADA
CFMB000.ADA	CFMB001.ADA	CFMB002.ADA	CFMB003.ADA	CFMB004.ADA	CFMB005.ADA
CFMB010.ADA	CFMB011.ADA	CFMB012.ADA	CFMB013.ADA	CFMB014.ADA	CFMB015.ADA
CFMB016.ADA	CFMB017.ADA	CFMB018.ADA	CFMB019.ADA	CFMB020.ADA	CFMB029.ADA
CFMB030.ADA	CFMB031.ADA	CFMB032.ADA	CFMB033.ADA	CFMB034.ADA	CFMB035.ADA
CFMB036.ADA	CFMB041.ADA	CFMB042.ADA	CFMB043.ADA	CFMB044.ADA	CFN9300.ADA
CFN9301.ADA	CFN9302.ADA	CFN9303.ADA	CFN9510.ADA	CFN9511.ADA	CFN9610.ADA
CFN9611.ADA	CFN9612.ADA	CFN9613.ADA	CGD0000C.ADA	CGD0000D.ADA	CGD0000E.ADA
CGD0001.ADA	CGD0002.ADA	CGD0003.ADA	CGD0004.ADA	CGD0005.ADA	CGD0006.ADA
CGD0007.ADA	CGD0008.ADA	CGD0009.ADA	CGD000D.ADA	CGD000E.ADA	CGD0019.ADA
CGD0029.ADA	CGD0039.ADA	CGD0049.ADA	CGN000F.ADA	CGN000G.ADA	CGN000H.ADA
CGN000I.ADA	CGN000J.ADA	CGN000K.ADA	CL09000.ADA	CL09100.ADA	CL09101.ADA
CL09111.ADA	CL09121.ADA	CL09131.ADA	CL09000.ADA	CLD9200.ADA	CLD9202.ADA
CLD9203.ADA	CLD9212.ADA	CLD9213.ADA	CLD9222.ADA	CLD9223.ADA	CLD9232.ADA
CLD9233.ADA	CLDE000.ADA	CLDE2B1.ADA	CLDE2B2.ADA	CLDE2C1.ADA	CLDE2C2.ADA
CLDE2C5.ADA	CLDE2C6.ADA	CLDE4B1.ADA	CLDE4B2.ADA	CLDE4C1.ADA	CLDE4C2.ADA
CLDE4C5.ADA	CLDE4C6.ADA	C000001.ADA	C000002.ADA	C000102.ADA	C000103.ADA
C000104.ADA	C000107.ADA	C000108.ADA	C000109.ADA	C000112.ADA	C000113.ADA
C000114.ADA	C000202.ADA	C000203.ADA	C000204.ADA	C000207.ADA	C000208.ADA
C000209.ADA	C000300.ADA	C000305.ADA	C000310.ADA	C000311.ADA	C000312.ADA
C000313.ADA	C000314.ADA	C000315.ADA	C000316.ADA	C000317.ADA	C000318.ADA
C000319.ADA	C000320.ADA	C000321.ADA	C000322.ADA	C000323.ADA	C000324.ADA
C000325.ADA	C000326.ADA	C000327.ADA	C000328.ADA	C000329.ADA	C000330.ADA
C000331.ADA	C000332.ADA	C000333.ADA	C000334.ADA	C000335.ADA	C000336.ADA
C000337.ADA	C000338.ADA	C000339.ADA	C000340.ADA	C000341.ADA	C000342.ADA
C000343.ADA	C000347.ADA	C000348.ADA	C000349.ADA	C000350.ADA	C000351.ADA
C000352.ADA	C000353.ADA	C000354.ADA	C000355.ADA	C000356.ADA	C000357.ADA
C000358.ADA	C000359.ADA	C000360.ADA	C000361.ADA	C000362.ADA	C000363.ADA
C000364.ADA	C000365.ADA	C000366.ADA	C000367.ADA	C000368.ADA	C000369.ADA
C00036A.ADA	C00036B.ADA	C00036F.ADA	C00036G.ADA	C00036H.ADA	C00036I.ADA
C00036J.ADA	C00036K.ADA	C00036L.ADA	C00036M.ADA	C00036N.ADA	C00036O.ADA
C00036V.ADA	C00036W.ADA	C000374.ADA	C000375.ADA	C000376.ADA	C000377.ADA
C000378.ADA	C000390.ADA	C000391.ADA	C000402.ADA	C000403.ADA	C000404.ADA
C000500.ADA	C000502.ADA	C000503.ADA	C000504.ADA	C000507.ADA	C000508.ADA
C000509.ADA	C000512.ADA	C000513.ADA	C000514.ADA	C000517.ADA	C000518.ADA
C000519.ADA	C000602.ADA	C000603.ADA	C000604.ADA	C000605.ADA	C000606.ADA
C000607.ADA	C000702.ADA	C000703.ADA	C000704.ADA	C000705.ADA	C000706.ADA
C000707.ADA	C000708.ADA	C000709.ADA	C000710.ADA	C000711.ADA	C000713.ADA
C000714.ADA	C000715.ADA	C000716.ADA	C000717.ADA	C000718.ADA	C000719.ADA
C000720.ADA	C000721.ADA	C00P000.ADA	C0D0344.ADA	C0D0345.ADA	C0D0346.ADA
C0D036C.ADA	C0D036D.ADA	C0D036E.ADA	C0D0360.ADA	C0D036P.ADA	C0D036Q.ADA
C0D036R.ADA	C0D036S.ADA	C0D036T.ADA	C0D036X.ADA	C0D036Y.ADA	C0D036Z.ADA
C0D0370.ADA	C0D0371.ADA	C0D0372.ADA	C0D0373.ADA	C0D0380.ADA	C0D0381.ADA
C0D0382.ADA	C0D0383.ADA	C0D0523.ADA	C0D0524.ADA	C0D0525.ADA	C0D0526.ADA
C0D0527.ADA	C0D0528.ADA	C0D0529.ADA	EA00000.ADA	EA0P000.ADA	EF03500.ADA
EF03501.ADA	EF03502.ADA	EF03503.ADA	EF03504.ADA	EF03509.ADA	EF03510.ADA
EF03511.ADA	EF03512.ADA	EF03513.ADA	EF03514.ADA	EF03517.ADA	EF03518.ADA
EF03519.ADA	EF03550.ADA	EF03600.ADA	EF03601.ADA	EF03602.ADA	EF03603.ADA
EF03604.ADA	EF03605.ADA	EF03606.ADA	EF03607.ADA	EF03608.ADA	EF03609.ADA
EF03610.ADA	EF03611.ADA	EF03612.ADA	EF03613.ADA	EF03614.ADA	EF03615.ADA
EF03616.ADA	EF03617.ADA	EF03618.ADA	EF03619.ADA	EF03620.ADA	EF03630.ADA
EF03631.ADA	EF03632.ADA	EF03633.ADA	EF03634.ADA	EF03635.ADA	EF03636.ADA
EF03637.ADA	EF03638.ADA	EF03639.ADA	EF03641.ADA	EF03642.ADA	EF03643.ADA
EF03644.ADA	EF03645.ADA	EF03646.ADA	EF03647.ADA	EF03648.ADA	EF03649.ADA
EF03650.ADA	EF03700.ADA	EF03701.ADA	EF03702.ADA	EF03703.ADA	EF03704.ADA
EF03800.ADA	EF03801.ADA	EF03802.ADA	EF03803.ADA	EF03804.ADA	EF03805.ADA
EF04120.ADA	EF04121.ADA	EF04122.ADA	EF04123.ADA	EF04124.ADA	EF04125.ADA
EF04126.ADA	EF04127.ADA	EF04129.ADA	EF0412B.ADA	EF04130.ADA	EF04131.ADA
EF04132.ADA	EF04133.ADA	EF0413D.ADA	EF0413H.ADA	EF0413I.ADA	EF0413J.ADA
EF0413K.ADA	EF04310.ADA	EF04311.ADA	EF04312.ADA	EF04510.ADA	EF04511.ADA
EF04512.ADA	EF04513.ADA	EF04514.ADA	EF04515.ADA	EF04516.ADA	EF04517.ADA
EF04518.ADA	EF04519.ADA	EF0451A.ADA	EF0451B.ADA	EF0451C.ADA	EF0451D.ADA
EF0451E.ADA	EF0451F.ADA	EF0451G.ADA	EF0451H.ADA	EF0451I.ADA	EF0451J.ADA
EF0451V.ADA	EF0451W.ADA	EF0451X.ADA	EF0451Y.ADA	EF0451Z.ADA	EF04520.ADA
EF04521.ADA	EF04522.ADA	EF04527.ADA	EF04529.ADA	EF04530.ADA	EF04531.ADA
EF04532.ADA	EF04533.ADA	EF04534.ADA	EF04535.ADA	EF04536.ADA	EF04537.ADA
EF04538.ADA	EF04539.ADA	EF0453A.ADA	EF0453B.ADA	EF0453C.ADA	EF04540.ADA
EF04541.ADA	EF04550.ADA	EF04551.ADA	EF04552.ADA	EF04553.ADA	EF04554.ADA

Table A-2. ACPS ANSI Tape Contents Volume ACPS02 (Continued)

EF04555.ADA	EF04556.ADA	EF04557.ADA	EF04558.ADA	EF04559.ADA	EF0455N.ADA
EF04550.ADA	EF04560.ADA	EF04562.ADA	EF04563.ADA	EF04600.ADA	EF04601.ADA
EF04602.ADA	EF0460E.ADA	EF0460F.ADA	EF0460G.ADA	EF0460H.ADA	EF0460I.ADA
EF0460J.ADA	EF0460M.ADA	EF05200.ADA	EF05201.ADA	EF05202.ADA	EF05203.ADA
EF05204.ADA	EF0520S.ADA	EF05206.ADA	EF05207.ADA	EF05208.ADA	EF05209.ADA
EF0520A.ADA	EF0520B.ADA	EF0520M.ADA	EF0520O.ADA	EF0520R.ADA	EF0520S.ADA
EF05210.ADA	EF05211.ADA	EF05212.ADA	EF05213.ADA	EF05214.ADA	EF0521S.ADA
EF0521H.ADA	EF0521I.ADA	EF0521J.ADA	EF0521K.ADA	EF0521L.ADA	EF0521N.ADA
EF05300.ADA	EF05301.ADA	EF05302.ADA	EF05303.ADA	EF05304.ADA	EF0530S.ADA
EF05306.ADA	EF05307.ADA	EF05308.ADA	EF05400.ADA	EF05401.ADA	EF05402.ADA
EF05403.ADA	EF05404.ADA	EF0540S.ADA	EF05406.ADA	EF05407.ADA	EF05408.ADA
EF05501.ADA	EF05502.ADA	EF05503.ADA	EF05504.ADA	EF0550S.ADA	EF05506.ADA
EF05507.ADA	EF05508.ADA	EF05509.ADA	EF0550A.ADA	EF0550B.ADA	EF0550C.ADA
EF0550D.ADA	EF0550E.ADA	EF0550F.ADA	EF06001.ADA	EF06009.ADA	EF06010.ADA
EF06011.ADA	EF06013.ADA	EF06014.ADA	EF0601S.ADA	EF06016.ADA	EF06017.ADA
EF06018.ADA	EF06019.ADA	EF06022.ADA	EF06023.ADA	EF06024.ADA	EF0602S.ADA
EF06026.ADA	EF06027.ADA	EF06028.ADA	EF06029.ADA	EF06030.ADA	EF06031.ADA
EF06032.ADA	EF06033.ADA	EF06043.ADA	EF06044.ADA	EF0604S.ADA	EF06046.ADA
EF06047.ADA	EF06048.ADA	EF06049.ADA	EF06050.ADA	EF06051.ADA	EF06052.ADA
EF06053.ADA	EF06060.ADA	EF06061.ADA	EF06062.ADA	EF06063.ADA	EF06064.ADA
EF0606S.ADA	EF06066.ADA	EF06067.ADA	EF06068.ADA	EF06069.ADA	EF06070.ADA
EF06071.ADA	EF06072.ADA	EF06073.ADA	EF06074.ADA	EF0607S.ADA	EF06076.ADA
EF06077.ADA	EF06078.ADA	EF06079.ADA	EF06100.ADA	EF06101.ADA	EF06108.ADA
EF06109.ADA	EF06110.ADA	EF06111.ADA	EF06112.ADA	EF06113.ADA	EF06114.ADA
EF0611S.ADA	EF06116.ADA	EF06117.ADA	EF06118.ADA	EF06119.ADA	EF06122.ADA
EF06132.ADA	EF06133.ADA	EF06134.ADA	EF0613S.ADA	EF06136.ADA	EF06137.ADA
EF06138.ADA	EF06139.ADA	EF06140.ADA	EF06142.ADA	EF06143.ADA	EF06144.ADA
EF0614S.ADA	EF06146.ADA	EF06147.ADA	EF06148.ADA	EF06149.ADA	EF06150.ADA
EF06152.ADA	EF06153.ADA	EF06154.ADA	EF0615S.ADA	EF06156.ADA	EF06157.ADA
EF06158.ADA	EF06159.ADA	EF06160.ADA	EF06162.ADA	EF06163.ADA	EF06164.ADA
EF0616S.ADA	EF06166.ADA	EF06167.ADA	EF06168.ADA	EF06169.ADA	EF06170.ADA
EF06172.ADA	EF06173.ADA	EF06174.ADA	EF0617S.ADA	EF06176.ADA	EF06177.ADA
EF06178.ADA	EF06179.ADA	EF06180.ADA	EF06182.ADA	EF06183.ADA	EF06184.ADA
EF0618S.ADA	EF06186.ADA	EF06187.ADA	EF06188.ADA	EF06189.ADA	EF06190.ADA
EF06191.ADA	EF06192.ADA	EF06193.ADA	EF06194.ADA	EF0619S.ADA	EF06196.ADA
EF06197.ADA	EF06198.ADA	EF06199.ADA	EF0619A.ADA	EF0619B.ADA	EF0619C.ADA
EF0619D.ADA	EF0619E.ADA	EF0619F.ADA	EF0619G.ADA	EF0619H.ADA	EF0619I.ADA
EF0619J.ADA	EF0619K.ADA	EF0619L.ADA	EF0619M.ADA	EF0619N.ADA	EF0619O.ADA
EF0619P.ADA	EF0619Q.ADA	EF0619R.ADA	EF06422.ADA	EF06423.ADA	EF06424.ADA
EF0642S.ADA	EF06426.ADA	EF06427.ADA	EF06428.ADA	EF06429.ADA	EF0642A.ADA
EF0642B.ADA	EF06802.ADA	EF06803.ADA	EF06806.ADA	EF06807.ADA	EF06808.ADA
EF06809.ADA	EF0680A.ADA	EF0680B.ADA	EF0680C.ADA	EF0680D.ADA	EF0680E.ADA
EF0680F.ADA	EF06810.ADA	EF06811.ADA	EF06814.ADA	EF0681S.ADA	EF06816.ADA
EF06817.ADA	EF06818.ADA	EF06819.ADA	EF06820.ADA	EF06821.ADA	EF06822.ADA
EF06823.ADA	EF06824.ADA	EF0682S.ADA	EF09500.ADA	EF09501.ADA	EF09502.ADA
EF09503.ADA	EF09504.ADA	EF0950S.ADA	EF09506.ADA	EF09507.ADA	EF09508.ADA
EF09509.ADA	EF09600.ADA	EF09601.ADA	EF09602.ADA	EF09603.ADA	EF09604.ADA
EF0960S.ADA	EF09606.ADA	EF09607.ADA	EF09710.ADA	EF09711.ADA	EF09712.ADA
EF09713.ADA	EF09720.ADA	EF09721.ADA	EF09731.ADA	EF09900.ADA	EF09901.ADA
EF09902.ADA	EF09903.ADA	EF09B01.ADA	EF0C100.ADA	EF0C101.ADA	EF0C102.A

Table A-2. ACPS ANSI Tape Contents Volume ACPS02 (Continued)

[illegible]

Table A-2. ACPS ANSI Tape Contents Volume ACPS02 (Concluded)

EFD6253.ADA	EFD6260.ADA	EFD6261.ADA	EFD6262.ADA	EFD6263.ADA	EFD6264.ADA
EFD6265.ADA	EFD6266.ADA	EFD6267.ADA	EFD6268.ADA	EFD6269.ADA	EFD6270.ADA
EFD6271.ADA	EFD6272.ADA	EFD6273.ADA	EFD6274.ADA	EFD6275.ADA	EFD6276.ADA
EFD6277.ADA	EFD6278.ADA	EFD6279.ADA	EFD6300.ADA	EFD6301.ADA	EFD6308.ADA
EFD6309.ADA	EFD6310.ADA	EFD6311.ADA	EFD6312.ADA	EFD6313.ADA	EFD6314.ADA
EFD6315.ADA	EFD6316.ADA	EFD6317.ADA	EFD6318.ADA	EFD6319.ADA	EFD6322.ADA
EFD6332.ADA	EFD6333.ADA	EFD6334.ADA	EFD6335.ADA	EFD6336.ADA	EFD6337.ADA
EFD6338.ADA	EFD6339.ADA	EFD6340.ADA	EFD6342.ADA	EFD6343.ADA	EFD6344.ADA
EFD6345.ADA	EFD6346.ADA	EFD6347.ADA	EFD6348.ADA	EFD6349.ADA	EFD6350.ADA
EFD6352.ADA	EFD6353.ADA	EFD6354.ADA	EFD6355.ADA	EFD6356.ADA	EFD6357.ADA
EFD6358.ADA	EFD6359.ADA	EFD6360.ADA	EFD6362.ADA	EFD6363.ADA	EFD6364.ADA
EFD6365.ADA	EFD6366.ADA	EFD6367.ADA	EFD6368.ADA	EFD6369.ADA	EFD6370.ADA
EFD6372.ADA	EFD6373.ADA	EFD6374.ADA	EFD6375.ADA	EFD6376.ADA	EFD6377.ADA
EFD6378.ADA	EFD6379.ADA	EFD6380.ADA	EFD6382.ADA	EFD6383.ADA	EFD6384.ADA
EFD6385.ADA	EFD6386.ADA	EFD6387.ADA	EFD6388.ADA	EFD6389.ADA	EFD6390.ADA
EFD680G.ADA	EFD680H.ADA	EFD680I.ADA	EFD680J.ADA	EFD9000.ADA	EFD9200.ADA
EFD9C00.ADA	EFD9C01.ADA	EFD9C02.ADA	EFDC104.ADA	EFDC105.ADA	EFDC106.ADA
EFDC107.ADA	EFDC310.ADA	EFDC311.ADA	EFDC313.ADA	EFDC314.ADA	EFDD600.ADA
EFDD601.ADA	EFDD602.ADA	EFDD603.ADA	EFDD604.ADA	EFDD605.ADA	EFDD606.ADA
EFDD607.ADA					

Total of 1369 files.

Figure A-3. ACPS ANSI Tape Contents Volume ACPS03

EFDD609.ADA	EFDD610.ADA	EFDD611.ADA	EFDDA01.ADA	EFDDA02.ADA	EFDE000.ADA
EFDE201.ADA	EFDE202.ADA	EFDE203.ADA	EFDE204.ADA	EFDE205.ADA	EFDE206.ADA
EFDE207.ADA	EFDE208.ADA	EFDE209.ADA	EFDE20A.ADA	EFDE211.ADA	EFDE212.ADA
EFDE213.ADA	EFDE214.ADA	EFDE215.ADA	EFDE216.ADA	EFDE220.ADA	EFDE231.ADA
EFDE232.ADA	EFDE233.ADA	EFDE234.ADA	EFDE235.ADA	EFDE236.ADA	EFDE237.ADA
EFDE238.ADA	EFDE239.ADA	EFDE23A.ADA	EFDE241.ADA	EFDE242.ADA	EFDE243.ADA
EFDE244.ADA	EFDE245.ADA	EFDE246.ADA	EFDE250.ADA	EFDE401.ADA	EFDE402.ADA
EFDE403.ADA	EFDE404.ADA	EFDE405.ADA	EFDE406.ADA	EFDE407.ADA	EFDE408.ADA
EFDE409.ADA	EFDE40A.ADA	EFDE411.ADA	EFDE412.ADA	EFDE413.ADA	EFDE414.ADA
EFDE415.ADA	EFDE416.ADA	EFDE420.ADA	EFDE431.ADA	EFDE432.ADA	EFDE433.ADA
EFDE434.ADA	EFDE435.ADA	EFDE436.ADA	EFDE437.ADA	EFDE438.ADA	EFDE439.ADA
EFDE43A.ADA	EFDE441.ADA	EFDE442.ADA	EFDE443.ADA	EFDE444.ADA	EFDE445.ADA
EFDE446.ADA	EFDE450.ADA	EFDF000.ADA	EFDF001.ADA	EFDF002.ADA	EFDF003.ADA
EFDF004.ADA	EFDF005.ADA	EFM9A00.ADA	EFM9A01.ADA	EFM9A02.ADA	EFM9A03.ADA
EFN9300.ADA	EFN9301.ADA	EFN9302.ADA	EFN9303.ADA	EFN9510.ADA	EFN9511.ADA
EFN9610.ADA	EFN9611.ADA	EFN9612.ADA	EFN9613.ADA	EG0000C.ADA	EG0P000.ADA
EGD0000.ADA	EGD0001.ADA	EGD0002.ADA	EGD0003.ADA	EGD0004.ADA	EGD0005.ADA
EGD0006.ADA	EGD0007.ADA	EGD0008.ADA	EGD0009.ADA	EGD000D.ADA	EGD000E.ADA
EGD0019.ADA	EGD0029.ADA	EGD0039.ADA	EGD0049.ADA	EGN000F.ADA	EGN000G.ADA
EGN000H.ADA	EGN000I.ADA	EGN000J.ADA	EGN000K.ADA	EL09000.ADA	EL09100.ADA
EL09101.ADA	EL09111.ADA	EL09121.ADA	EL09131.ADA	EL0P000.ADA	ELD9200.ADA
ELD9202.ADA	ELD9203.ADA	ELD9212.ADA	ELD9213.ADA	ELD9222.ADA	ELD9223.ADA
ELD9232.ADA	ELD9233.ADA	ELDE000.ADA	ELDE2B1.ADA	ELDE2B2.ADA	ELDE2C1.ADA
ELDE2C2.ADA	ELDE2C5.ADA	ELDE2C6.ADA	ELDE4B1.ADA	ELDE4B2.ADA	ELDE4C1.ADA
ELDE4C2.ADA	ELDE4C5.ADA	ELDE4C6.ADA	E000001.ADA	E000002.ADA	E000102.ADA
E000103.ADA	E000104.ADA	E000107.ADA	E000108.ADA	E000109.ADA	E000112.ADA
E000113.ADA	E000114.ADA	E000202.ADA	E000203.ADA	E000204.ADA	E000207.ADA
E000208.ADA	E000209.ADA	E000305.ADA	E000305.ADA	E000310.ADA	E000311.ADA
E000312.ADA	E000313.ADA	E000314.ADA	E000315.ADA	E000316.ADA	E000317.ADA
E000318.ADA	E000319.ADA	E000320.ADA	E000321.ADA	E000322.ADA	E000323.ADA
E000324.ADA	E000325.ADA	E000326.ADA	E000327.ADA	E000328.ADA	E000329.ADA
E000330.ADA	E000331.ADA	E000332.ADA	E000333.ADA	E000334.ADA	E000335.ADA
E000336.ADA	E000337.ADA	E000338.ADA	E000339.ADA	E000340.ADA	E000341.ADA
E000342.ADA	E000343.ADA	E000347.ADA	E000348.ADA	E000349.ADA	E000350.ADA
E000351.ADA	E000352.ADA	E000353.ADA	E000354.ADA	E000355.ADA	E000356.ADA
E000357.ADA	E000358.ADA	E000359.ADA	E000360.ADA	E000361.ADA	E000362.ADA
E000363.ADA	E000364.ADA	E000365.ADA	E000366.ADA	E000367.ADA	E000368.ADA
E000369.ADA	E00036A.ADA	E00036B.ADA	E00036F.ADA	E00036G.ADA	E00036H.ADA
E00036I.ADA	E00036J.ADA	E00036K.ADA	E00036L.ADA	E00036M.ADA	E00036N.ADA
E00036U.ADA	E00036V.ADA	E00036W.ADA	E000374.ADA	E000375.ADA	E000376.ADA
E000377.ADA	E000378.ADA	E000390.ADA	E000391.ADA	E000402.ADA	E000403.ADA
E000404.ADA	E000500.ADA	E000502.ADA	E000503.ADA	E000504.ADA	E000507.ADA
E000508.ADA	E000509.ADA	E000512.ADA	E000513.ADA	E000514.ADA	E000517.ADA
E000518.ADA	E000519.ADA	E000602.ADA	E000603.ADA	E000604.ADA	E000605.ADA
E000606.ADA	E000607.ADA	E000702.ADA	E000703.ADA	E000704.ADA	E000705.ADA
E000706.ADA	E000707.ADA	E000708.ADA	E000709.ADA	E000710.ADA	E000711.ADA
E000713.ADA	E000714.ADA	E000715.ADA	E000716.ADA	E000717.ADA	E000718.ADA
E000719.ADA	E000720.ADA	E000721.ADA	E00P000.ADA	E0D0344.ADA	E0D0345.ADA
E0D0346.ADA	E0D036C.ADA	E0D036D.ADA	E0D036E.ADA	E0D036D.ADA	E0D036P.ADA
E0D036Q.ADA	E0D036R.ADA	E0D036S.ADA	E0D036T.ADA	E0D036X.ADA	E0D036Y.ADA
E0D036Z.ADA	E0D0370.ADA	E0D0371.ADA	E0D0372.ADA	E0D0373.ADA	E0D0380.ADA
E0D0381.ADA	E0D0382.ADA	E0D0383.ADA	E0D0523.ADA	E0D0524.ADA	E0D0525.ADA
E0D0526.ADA	E0D0527.ADA	E0D0528.ADA	E0D0529.ADA	OURDMP.ADA	OURSYS.ADA
OURSYSR.ADA	SA00000.ADA	SA0P000.ADA	SF03500.ADA	SF03501.ADA	SF03502.ADA
SF03503.ADA	SF03504.ADA	SF03509.ADA	SF03510.ADA	SF03511.ADA	SF03512.ADA
SF03513.ADA	SF03514.ADA	SF03517.ADA	SF03518.ADA	SF03519.ADA	SF03550.ADA
SF03600.ADA	SF03601.ADA	SF03602.ADA	SF03603.ADA	SF03604.ADA	SF03605.ADA
SF03606.ADA	SF03607.ADA	SF03608.ADA	SF03609.ADA	SF03610.ADA	SF03611.ADA
SF03612.ADA	SF03613.ADA	SF03614.ADA	SF03615.ADA	SF03616.ADA	SF03617.ADA
SF03618.ADA	SF03619.ADA	SF03620.ADA	SF03630.ADA	SF03631.ADA	SF03632.ADA
SF03633.ADA	SF03634.ADA	SF03635.ADA	SF03636.ADA	SF03637.ADA	SF03638.ADA
SF03639.ADA	SF03641.ADA	SF03642.ADA	SF03643.ADA	SF03644.ADA	SF03645.ADA
SF03646.ADA	SF03647.ADA	SF03648.ADA	SF03649.ADA	SF03650.ADA	SF03700.ADA
SF03701.ADA	SF03702.ADA	SF03703.ADA	SF03704.ADA	SF03800.ADA	SF03801.ADA
SF03802.ADA	SF03803.ADA	SF03804.ADA	SF03805.ADA	SF04120.ADA	SF04121.ADA
SF04122.ADA	SF04123.ADA	SF04124.ADA	SF04125.ADA	SF04126.ADA	SF04127.ADA
SF04129.ADA	SF0412B.ADA	SF04130.ADA	SF04131.ADA	SF04132.ADA	SF04133.ADA
SF0413D.ADA	SF0413H.ADA	SF0413I.ADA	SF0413J.ADA	SF0413K.ADA	SF04310.ADA
SF04311.ADA	SF04312.ADA	SF04510.ADA	SF04511.ADA	SF04512.ADA	SF04513.ADA

Figure A-3. ACPS ANSI Tape Contents Volume ACPS03 (Continued)

[illegible]

Figure A-3. ACPS ANSI Tape Contents Volume ACPS03 (Continued)

SFD4804.ADA	SFD520C.ADA	SFD520D.ADA	SFD520E.ADA	SFD520F.ADA	SFD520G.ADA
SFD520H.ADA	SFD520I.ADA	SFD520J.ADA	SFD520N.ADA	SFD520P.ADA	SFD520Q.ADA
SFD5218.ADA	SFD5219.ADA	SFD521A.ADA	SFD521B.ADA	SFD521C.ADA	SFD521D.ADA
SFD521E.ADA	SFD521F.ADA	SFD521K.ADA	SFD521M.ADA	SFD521O.ADA	SFD521P.ADA
SFD6201.ADA	SFD6209.ADA	SFD6210.ADA	SFD6211.ADA	SFD6213.ADA	SFD6214.ADA
SFD6215.ADA	SFD6216.ADA	SFD6217.ADA	SFD6218.ADA	SFD6219.ADA	SFD6222.ADA
SFD6223.ADA	SFD6224.ADA	SFD6225.ADA	SFD6226.ADA	SFD6227.ADA	SFD6228.ADA
SFD6229.ADA	SFD6230.ADA	SFD6231.ADA	SFD6232.ADA	SFD6233.ADA	SFD6243.ADA
SFD6244.ADA	SFD6245.ADA	SFD6246.ADA	SFD6247.ADA	SFD6248.ADA	SFD6249.ADA
SFD6250.ADA	SFD6251.ADA	SFD6252.ADA	SFD6253.ADA	SFD6260.ADA	SFD6261.ADA
SFD6262.ADA	SFD6263.ADA	SFD6264.ADA	SFD6265.ADA	SFD6266.ADA	SFD6267.ADA
SFD6268.ADA	SFD6269.ADA	SFD6270.ADA	SFD6271.ADA	SFD6272.ADA	SFD6273.ADA
SFD6274.ADA	SFD6275.ADA	SFD6276.ADA	SFD6277.ADA	SFD6278.ADA	SFD6279.ADA
SFD6300.ADA	SFD6301.ADA	SFD6308.ADA	SFD6309.ADA	SFD6310.ADA	SFD6311.ADA
SFD6312.ADA	SFD6313.ADA	SFD6314.ADA	SFD6315.ADA	SFD6316.ADA	SFD6317.ADA
SFD6318.ADA	SFD6319.ADA	SFD6322.ADA	SFD6332.ADA	SFD6333.ADA	SFD6334.ADA
SFD6335.ADA	SFD6336.ADA	SFD6337.ADA	SFD6338.ADA	SFD6339.ADA	SFD6340.ADA
SFD6342.ADA	SFD6343.ADA	SFD6344.ADA	SFD6345.ADA	SFD6346.ADA	SFD6347.ADA
SFD6348.ADA	SFD6349.ADA	SFD6350.ADA	SFD6352.ADA	SFD6353.ADA	SFD6354.ADA
SFD6355.ADA	SFD6356.ADA	SFD6357.ADA	SFD6358.ADA	SFD6359.ADA	SFD6360.ADA
SFD6362.ADA	SFD6363.ADA	SFD6364.ADA	SFD6365.ADA	SFD6366.ADA	SFD6367.ADA
SFD6368.ADA	SFD6369.ADA	SFD6370.ADA	SFD6372.ADA	SFD6373.ADA	SFD6374.ADA
SFD6375.ADA	SFD6376.ADA	SFD6377.ADA	SFD6378.ADA	SFD6379.ADA	SFD6380.ADA
SFD6382.ADA	SFD6383.ADA	SFD6384.ADA	SFD6385.ADA	SFD6386.ADA	SFD6387.ADA
SFD6388.ADA	SFD6389.ADA	SFD6390.ADA	SFD680G.ADA	SFD680H.ADA	SFD680I.ADA
SFD680J.ADA	SFD9000.ADA	SFD9200.ADA	SFD9C00.ADA	SFD9C01.ADA	SFD9C02.ADA
SFDC104.ADA	SFDC105.ADA	SFDC106.ADA	SFDC107.ADA	SFDC310.ADA	SFDC311.ADA
SFDC313.ADA	SFDC314.ADA	SFDD600.ADA	SFDD601.ADA	SFDD602.ADA	SFDD603.ADA
SFDD604.ADA	SFDD605.ADA	SFDD606.ADA	SFDD607.ADA	SFDD609.ADA	SFDD610.ADA
SFDD611.ADA	SFDD722.ADA	SFDD723.ADA	SFDD724.ADA	SFDD725.ADA	SFDD72A.ADA
SFDD72B.ADA	SFDD72C.ADA	SFDD72E.ADA	SFDD72F.ADA	SFDD72G.ADA	SFDD72H.ADA
SFDD72I.ADA	SFDDA01.ADA	SFDDA02.ADA	SFDE000.ADA	SFDE201.ADA	SFDE202.ADA
SFDE203.ADA	SFDE204.ADA	SFDE205.ADA	SFDE206.ADA	SFDE207.ADA	SFDE208.ADA
SFDE209.ADA	SFDE20A.ADA	SFDE211.ADA	SFDE212.ADA	SFDE213.ADA	SFDE214.ADA
SFDE215.ADA	SFDE216.ADA	SFDE220.ADA	SFDE231.ADA	SFDE232.ADA	SFDE233.ADA
SFDE234.ADA	SFDE235.ADA	SFDE236.ADA	SFDE237.ADA	SFDE238.ADA	SFDE239.ADA
SFDE23A.ADA	SFDE241.ADA	SFDE242.ADA	SFDE243.ADA	SFDE244.ADA	SFDE245.ADA
SFDE246.ADA	SFDE250.ADA	SFDE401.ADA	SFDE402.ADA	SFDE403.ADA	SFDE404.ADA
SFDE405.ADA	SFDE406.ADA	SFDE407.ADA	SFDE408.ADA	SFDE409.ADA	SFDE40A.ADA
SFDE411.ADA	SFDE412.ADA	SFDE413.ADA	SFDE414.ADA	SFDE415.ADA	SFDE416.ADA
SFDE420.ADA	SFDE431.ADA	SFDE432.ADA	SFDE433.ADA	SFDE434.ADA	SFDE435.ADA
SFDE436.ADA	SFDE437.ADA	SFDE438.ADA	SFDE439.ADA	SFDE43A.ADA	SFDE441.ADA
SFDE442.ADA	SFDE443.ADA	SFDE444.ADA	SFDE445.ADA	SFDE446.ADA	SFDE450.ADA
SFDF000.ADA	SFDF001.ADA	SFDF002.ADA	SFDF003.ADA	SFDF004.ADA	SFDF005.ADA
SFM9A00.ADA	SFM9A01.ADA	SFM9A02.ADA	SFM9A03.ADA	SFMB000.ADA	SFMB001.ADA
SFMB002.ADA	SFMB003.ADA	SFMB004.ADA	SFMB005.ADA	SFMB006.ADA	SFMB007.ADA
SFMB008.ADA	SFMB009.ADA	SFMB010.ADA	SFMB011.ADA	SFMB012.ADA	SFMB013.ADA
SFMB014.ADA	SFMB015.ADA	SFMB016.ADA	SFMB017.ADA	SFMB018.ADA	SFMB019.ADA
SFMB020.ADA	SFMB021.ADA	SFMB022.ADA	SFMB023.ADA	SFMB024.ADA	SFMB025.ADA
SFMB026.ADA	SFMB027.ADA	SFMB028.ADA	SFMB029.ADA	SFMB030.ADA	SFMB031.ADA
SFMB032.ADA	SFMB033.ADA	SFMB034.ADA	SFMB035.ADA	SFMB036.ADA	SFMB037.ADA
SFMB038.ADA	SFMB039.ADA	SFMB040.ADA	SFMB041.ADA	SFMB042.ADA	SFMB043.ADA
SFMB044.ADA	SFN9300.ADA	SFN9301.ADA	SFN9302.ADA	SFN9303.ADA	SFN9510.ADA
SFN9511.ADA	SFN9610.ADA	SFN9611.ADA	SFN9612.ADA	SFN9613.ADA	SG0000C.ADA
SG0P000.ADA	SGD0000.ADA	SGD0001.ADA	SGD0002.ADA	SGD0003.ADA	SGD0004.ADA
SGD0005.ADA	SGD0006.ADA	SGD0007.ADA	SGD0008.ADA	SGD0009.ADA	SGD000D.ADA
SGD000E.ADA	SGD0019.ADA	SGD0029.ADA	SGD0039.ADA	SGD0049.ADA	SGN000F.ADA
SGN000G.ADA	SGN000H.ADA	SGN000I.ADA	SGN000J.ADA	SGN000K.ADA	SL09000.ADA
SL09100.ADA	SL09101.ADA	SL09111.ADA	SL09121.ADA	SL09131.ADA	SL0P000.ADA
SLD9200.ADA	SLD9202.ADA	SLD9203.ADA	SLD9212.ADA	SLD9213.ADA	SLD9222.ADA
SLD9223.ADA	SLD9232.ADA	SLD9233.ADA	SLDE000.ADA	SLDE2B1.ADA	SLDE2B2.ADA
SLDE2C1.ADA	SLDE2C2.ADA	SLDE2C5.ADA	SLDE2C6.ADA	SLDE4B1.ADA	SLDE4B2.ADA
SLDE4C1.ADA	SLDE4C2.ADA	SLDE4C5.ADA	SLDE4C6.ADA	S000001.ADA	S000002.ADA
S000102.ADA	S000103.ADA	S000104.ADA	S000107.ADA	S000108.ADA	S000109.ADA
S000112.ADA	S000113.ADA	S000114.ADA	S000202.ADA	S000203.ADA	S000204.ADA
S000207.ADA	S000208.ADA	S000209.ADA	S000300.ADA	S000305.ADA	S000310.ADA
S000311.ADA	S000312.ADA	S000313.ADA	S000314.ADA	S000315.ADA	S000316.ADA
S000317.ADA	S000318.ADA	S000319.ADA	S000320.ADA	S000321.ADA	S000322.ADA
S000323.ADA	S000324.ADA	S000325.ADA	S000326.ADA	S000327.ADA	S000328.ADA
S000329.ADA	S000330.ADA	S000331.ADA	S000332.ADA	S000333.ADA	S000334.ADA

Figure A-3. ACPS ANSI Tape Contents Volume ACPS03 (Concluded)

S000335.ADA	S000336.ADA	S000337.ADA	S000338.ADA	S000339.ADA	S000340.ADA
S000341.ADA	S000342.ADA	S000343.ADA	S000347.ADA	S000348.ADA	S000349.ADA
S000350.ADA	S000351.ADA	S000352.ADA	S000353.ADA	S000354.ADA	S000355.ADA
S000356.ADA	S000357.ADA	S000358.ADA	S000359.ADA	S000360.ADA	S000361.ADA
S000362.ADA	S000363.ADA	S000364.ADA	S000365.ADA	S000366.ADA	S000367.ADA
S000368.ADA	S000369.ADA	S00036A.ADA	S00036B.ADA	S00036F.ADA	S00036G.ADA
S00036H.ADA	S00036I.ADA	S00036J.ADA	S00036K.ADA	S00036L.ADA	S00036M.ADA
S00036N.ADA	S00036U.ADA	S00036V.ADA	S00036W.ADA	S000374.ADA	S000375.ADA
S000376.ADA	S000377.ADA	S000378.ADA	S000390.ADA	S000391.ADA	S000402.ADA
S000403.ADA	S000404.ADA	S000500.ADA	S000502.ADA	S000503.ADA	S000504.ADA
S000507.ADA	S000508.ADA	S000509.ADA	S000512.ADA	S000513.ADA	S000514.ADA
S000517.ADA	S000518.ADA	S000519.ADA	S000602.ADA	S000603.ADA	S000604.ADA
S000605.ADA	S000606.ADA	S000607.ADA	S000702.ADA	S000703.ADA	S000704.ADA
S000705.ADA	S000706.ADA	S000707.ADA	S000708.ADA	S000709.ADA	S000710.ADA
S000711.ADA	S000713.ADA	S000714.ADA	S000715.ADA	S000716.ADA	S000717.ADA
S000718.ADA	S000719.ADA	S000720.ADA	S000721.ADA	S00P000.ADA	S0D0344.ADA
S0D0345.ADA	S0D0346.ADA	S0D036C.ADA	S0D036D.ADA	S0D036E.ADA	S0D0360.ADA
S0D036P.ADA	S0D036Q.ADA	S0D036R.ADA	S0D036S.ADA	S0D036T.ADA	S0D036X.ADA
S0D036Y.ADA	S0D036Z.ADA	S0D0370.ADA	S0D0371.ADA	S0D0372.ADA	S0D0373.ADA
S0D0380.ADA	S0D0381.ADA	S0D0382.ADA	S0D0383.ADA	S0D0523.ADA	S0D0524.ADA
S0D0525.ADA	S0D0526.ADA	S0D0527.ADA	S0D0528.ADA	S0D0529.ADA	TA00000.ADA
TA0P000.ADA	TF03500.ADA	TF03501.ADA	TF03502.ADA	TF03503.ADA	TF03504.ADA
TF03509.ADA	TF03510.ADA	TF03511.ADA	TF03512.ADA	TF03513.ADA	TF03514.ADA
TF03517.ADA	TF03518.ADA	TF03519.ADA	TF03550.ADA		

Total of 1396 files.

Figure A-4. ACPS ANSI Tape Contents Volume ACPS04

[illegible]

Figure A-4. ACPS ANSI Tape Contents Volume ACPS04 (Continued)

TF09721.ADA	TF09731.ADA	TF09900.ADA	TF09901.ADA	TF09902.ADA	TF09903.ADA
TF09801.ADA	TF0C100.ADA	TF0C101.ADA	TF0C102.ADA	TF0C103.ADA	TF0C300.ADA
TF0C301.ADA	TF0C302.ADA	TF0C303.ADA	TF0C304.ADA	TF0C305.ADA	TF0C306.ADA
TF0C307.ADA	TF0D720.ADA	TF0D721.ADA	TF0D727.ADA	TF0D728.ADA	TF0D729.ADA
TF0D72D.ADA	TF0P000.ADA	TFD3551.ADA	TFD412A.ADA	TFD412C.ADA	TFD412D.ADA
TFD4135.ADA	TFD4136.ADA	TFD4137.ADA	TFD4138.ADA	TFD4139.ADA	TFD413A.ADA
TFD413B.ADA	TFD413C.ADA	TFD413E.ADA	TFD413G.ADA	TFD451K.ADA	TFD451L.ADA
TFD451M.ADA	TFD451N.ADA	TFD4510.ADA	TFD451P.ADA	TFD451Q.ADA	TFD451R.ADA
TFD451S.ADA	TFD4523.ADA	TFD4524.ADA	TFD4525.ADA	TFD4526.ADA	TFD4528.ADA
TFD452A.ADA	TFD455D.ADA	TFD455E.ADA	TFD455F.ADA	TFD455G.ADA	TFD455I.ADA
TFD455J.ADA	TFD455K.ADA	TFD455L.ADA	TFD4566.ADA	TFD4567.ADA	TFD4568.ADA
TFD4569.ADA	TFD4603.ADA	TFD4604.ADA	TFD4605.ADA	TFD4606.ADA	TFD4607.ADA
TFD4608.ADA	TFD4609.ADA	TFD460A.ADA	TFD460B.ADA	TFD460K.ADA	TFD460L.ADA
TFD460N.ADA	TFD4600.ADA	TFD4800.ADA	TFD4801.ADA	TFD4802.ADA	TFD4803.ADA
TFD4804.ADA	TFD520C.ADA	TFD520D.ADA	TFD520E.ADA	TFD520F.ADA	TFD520G.ADA
TFD520H.ADA	TFD520I.ADA	TFD520J.ADA	TFD520N.ADA	TFD520P.ADA	TFD520Q.ADA
TFD5218.ADA	TFD5219.ADA	TFD521A.ADA	TFD521B.ADA	TFD521C.ADA	TFD521D.ADA
TFD521E.ADA	TFD521F.ADA	TFD521K.ADA	TFD521M.ADA	TFD5210.ADA	TFD521P.ADA
TFD6201.ADA	TFD6209.ADA	TFD6210.ADA	TFD6211.ADA	TFD6213.ADA	TFD6214.ADA
TFD6215.ADA	TFD6216.ADA	TFD6217.ADA	TFD6218.ADA	TFD6219.ADA	TFD6222.ADA
TFD6223.ADA	TFD6224.ADA	TFD6225.ADA	TFD6226.ADA	TFD6227.ADA	TFD6228.ADA
TFD6229.ADA	TFD6230.ADA	TFD6231.ADA	TFD6232.ADA	TFD6233.ADA	TFD6243.ADA
TFD6244.ADA	TFD6245.ADA	TFD6246.ADA	TFD6247.ADA	TFD6248.ADA	TFD6249.ADA
TFD6250.ADA	TFD6251.ADA	TFD6252.ADA	TFD6253.ADA	TFD6260.ADA	TFD6261.ADA
TFD6262.ADA	TFD6263.ADA	TFD6264.ADA	TFD6265.ADA	TFD6266.ADA	TFD6267.ADA
TFD6268.ADA	TFD6269.ADA	TFD6270.ADA	TFD6271.ADA	TFD6272.ADA	TFD6273.ADA
TFD6274.ADA	TFD6275.ADA	TFD6276.ADA	TFD6277.ADA	TFD6278.ADA	TFD6279.ADA
TFD6300.ADA	TFD6301.ADA	TFD6308.ADA	TFD6309.ADA	TFD6310.ADA	TFD6311.ADA
TFD6312.ADA	TFD6313.ADA	TFD6314.ADA	TFD6315.ADA	TFD6316.ADA	TFD6317.ADA
TFD6318.ADA	TFD6319.ADA	TFD6322.ADA	TFD6332.ADA	TFD6333.ADA	TFD6334.ADA
TFD6335.ADA	TFD6336.ADA	TFD6337.ADA	TFD6338.ADA	TFD6339.ADA	TFD6340.ADA
TFD6342.ADA	TFD6343.ADA	TFD6344.ADA	TFD6345.ADA	TFD6346.ADA	TFD6347.ADA
TFD6348.ADA	TFD6349.ADA	TFD6350.ADA	TFD6352.ADA	TFD6353.ADA	TFD6354.ADA
TFD6355.ADA	TFD6356.ADA	TFD6357.ADA	TFD6358.ADA	TFD6359.ADA	TFD6360.ADA
TFD6362.ADA	TFD6363.ADA	TFD6364.ADA	TFD6365.ADA	TFD6366.ADA	TFD6367.ADA
TFD6368.ADA	TFD6369.ADA	TFD6370.ADA	TFD6372.ADA	TFD6373.ADA	TFD6374.ADA
TFD6375.ADA	TFD6376.ADA	TFD6377.ADA	TFD6378.ADA	TFD6379.ADA	TFD6380.ADA
TFD6382.ADA	TFD6383.ADA	TFD6384.ADA	TFD6385.ADA	TFD6386.ADA	TFD6387.ADA
TFD6388.ADA	TFD6389.ADA	TFD6390.ADA	TFD680G.ADA	TFD680H.ADA	TFD680I.ADA
TFD680J.ADA	TFD9000.ADA	TFD9200.ADA	TFD9C00.ADA	TFD9C01.ADA	TFD9C02.ADA
TFDC104.ADA	TFDC105.ADA	TFDC106.ADA	TFDC107.ADA	TFDC310.ADA	TFDC311.ADA
TFDC313.ADA	TFDC314.ADA	TFDD600.ADA	TFDD601.ADA	TFDD602.ADA	TFDD603.ADA
TFDD604.ADA	TFDD605.ADA	TFDD606.ADA	TFDD607.ADA	TFDD609.ADA	TFDD610.ADA
TFDD611.ADA	TFDD722.ADA	TFDD723.ADA	TFDD724.ADA	TFDD725.ADA	TFDD72A.ADA
TFDD72B.ADA	TFDD72C.ADA	TFDD72E.ADA	TFDD72F.ADA	TFDD72G.ADA	TFDD72H.ADA
TFDD72I.ADA	TFDDA01.ADA	TFDDA02.ADA	TFDE000.ADA	TFDE201.ADA	TFDE202.ADA
TFDE203.ADA	TFDE204.ADA	TFDE205.ADA	TFDE206.ADA	TFDE207.ADA	TFDE208.ADA
TFDE209.ADA	TFDE20A.ADA	TFDE211.ADA	TFDE212.ADA	TFDE213.ADA	TFDE214.ADA
TFDE215.ADA	TFDE216.ADA	TFDE220.ADA	TFDE231.ADA	TFDE232.ADA	TFDE233.ADA
TFDE234.ADA	TFDE235.ADA	TFDE236.ADA	TFDE237.ADA	TFDE238.ADA	TFDE239.ADA
TFDE23A.ADA	TFDE241.ADA	TFDE242.ADA	TFDE243.ADA	TFDE244.ADA	TFDE245.ADA
TFDE246.ADA	TFDE250.ADA	TFDE401.ADA	TFDE402.ADA	TFDE403.ADA	TFDE404.ADA
TFDE405.ADA	TFDE406.ADA	TFDE407.ADA	TFDE408.ADA	TFDE409.ADA	TFDE40A.ADA
TFDE411.ADA	TFDE412.ADA	TFDE413.ADA	TFDE414.ADA	TFDE415.ADA	TFDE416.ADA
TFDE420.ADA	TFDE431.ADA	TFDE432.ADA	TFDE433.ADA	TFDE434.ADA	TFDE435.ADA
TFDE436.ADA	TFDE437.ADA	TFDE438.ADA	TFDE439.ADA	TFDE43A.ADA	TFDE441.ADA
TFDE442.ADA	TFDE443.ADA	TFDE444.ADA	TFDE445.ADA	TFDE446.ADA	TFDE450.ADA
TFDF000.ADA	TFDF001.ADA	TFDF002.ADA	TFDF003.ADA	TFDF004.ADA	TFDF005.ADA
TFM9A00.ADA	TFM9A01.ADA	TFM9A02.ADA	TFM9A03.ADA	TFMB000.ADA	TFMB001.ADA
TFMB002.ADA	TFMB003.ADA	TFMB004.ADA	TFMB005.ADA	TFMB006.ADA	TFMB007.ADA
TFMB008.ADA	TFMB009.ADA	TFMB010.ADA	TFMB011.ADA	TFMB012.ADA	TFMB013.ADA
TFMB014.ADA	TFMB015.ADA	TFMB016.ADA	TFMB017.ADA	TFMB018.ADA	TFMB019.ADA
TFMB020.ADA	TFMB021.ADA	TFMB022.ADA	TFMB023.ADA	TFMB024.ADA	TFMB025.ADA
TFMB026.ADA	TFMB027.ADA	TFMB028.ADA	TFMB029.ADA	TFMB030.ADA	TFMB031.ADA
TFMB032.ADA	TFMB033.ADA	TFMB034.ADA	TFMB035.ADA	TFMB036.ADA	TFMB037.ADA
TFMB038.ADA	TFMB039.ADA	TFMB040.ADA	TFMB041.ADA	TFMB042.ADA	TFMB043.ADA
TFMB044.ADA	TFN9300.ADA	TFN9301.ADA	TFN9302.ADA	TFN9303.ADA	TFN9510.ADA
TFN9511.ADA	TFN9610.ADA	TFN9611.ADA	TFN9612.ADA	TFN9613.ADA	TG0000C.ADA
TG0P000.ADA	TGD0000.ADA	TGD0001.ADA	TGD0002.ADA	TGD0003.ADA	TGD0004.ADA
TGD0005.ADA	TGD0006.ADA	TGD0007.ADA	TGD0008.ADA	TGD0009.ADA	TGD000D.ADA

Figure A-4. ACPS ANSI Tape Contents Volume ACPS04 (Continued)

TGD000E.ADA	TGD0019.ADA	TGD0029.ADA	TGD0039.ADA	TGD0049.ADA	TGN000F.ADA
TGN000G.ADA	TGN000H.ADA	TGN000I.ADA	TGN000J.ADA	TGN000K.ADA	TL09000.ADA
TL09100.ADA	TL09101.ADA	TL09111.ADA	TL09121.ADA	TL09131.ADA	TLOP000.ADA
TLD9200.ADA	TLD9202.ADA	TLD9203.ADA	TLD9212.ADA	TLD9213.ADA	TLD9222.ADA
TLD9223.ADA	TLD9232.ADA	TLD9233.ADA	TLDE000.ADA	TLDE2B1.ADA	TLDE2B2.ADA
TLDE2C1.ADA	TLDE2C2.ADA	TLDE2C5.ADA	TLDE2C6.ADA	TLDE4B1.ADA	TLDE4B2.ADA
TLDE4C1.ADA	TLDE4C2.ADA	TLDE4C5.ADA	TLDE4C6.ADA	T000001.ADA	T000002.ADA
T000102.ADA	T000103.ADA	T000104.ADA	T000107.ADA	T000108.ADA	T000109.ADA
T000112.ADA	T000113.ADA	T000114.ADA	T000202.ADA	T000203.ADA	T000204.ADA
T000207.ADA	T000208.ADA	T000209.ADA	T000300.ADA	T000305.ADA	T000310.ADA
T000311.ADA	T000312.ADA	T000313.ADA	T000314.ADA	T000315.ADA	T000316.ADA
T000317.ADA	T000318.ADA	T000319.ADA	T000320.ADA	T000321.ADA	T000322.ADA
T000323.ADA	T000324.ADA	T000325.ADA	T000326.ADA	T000327.ADA	T000328.ADA
T000329.ADA	T000330.ADA	T000331.ADA	T000332.ADA	T000333.ADA	T000334.ADA
T000335.ADA	T000336.ADA	T000337.ADA	T000338.ADA	T000339.ADA	T000340.ADA
T000341.ADA	T000342.ADA	T000343.ADA	T000347.ADA	T000348.ADA	T000349.ADA
T000350.ADA	T000351.ADA	T000352.ADA	T000353.ADA	T000354.ADA	T000355.ADA
T000356.ADA	T000357.ADA	T000358.ADA	T000359.ADA	T000360.ADA	T000361.ADA
T000362.ADA	T000363.ADA	T000364.ADA	T000365.ADA	T000366.ADA	T000367.ADA
T000368.ADA	T000369.ADA	T00036A.ADA	T00036B.ADA	T00036F.ADA	T00036G.ADA
T00036H.ADA	T00036I.ADA	T00036J.ADA	T00036K.ADA	T00036L.ADA	T00036M.ADA
T00036N.ADA	T00036U.ADA	T00036V.ADA	T00036W.ADA	T000374.ADA	T000375.ADA
T000376.ADA	T000377.ADA	T000378.ADA	T000390.ADA	T000391.ADA	T000402.ADA
T000403.ADA	T000404.ADA	T000500.ADA	T000502.ADA	T000503.ADA	T000504.ADA
T000507.ADA	T000508.ADA	T000509.ADA	T000512.ADA	T000513.ADA	T000514.ADA
T000517.ADA	T000518.ADA	T000519.ADA	T000602.ADA	T000603.ADA	T000604.ADA
T000605.ADA	T000606.ADA	T000607.ADA	T000702.ADA	T000703.ADA	T000704.ADA
T000705.ADA	T000706.ADA	T000707.ADA	T000708.ADA	T000709.ADA	T000710.ADA
T000711.ADA	T000713.ADA	T000714.ADA	T000715.ADA	T000716.ADA	T000717.ADA
T000718.ADA	T000719.ADA	T000720.ADA	T000721.ADA	T00P000.ADA	T000344.ADA
TOD0345.ADA	TOD0346.ADA	TOD036C.ADA	TOD036D.ADA	TOD036E.ADA	TOD0360.ADA
TOD036P.ADA	TOD036Q.ADA	TOD036R.ADA	TOD036S.ADA	TOD036T.ADA	TOD036X.ADA
TOD036Y.ADA	TOD036Z.ADA	TOD0370.ADA	TOD0371.ADA	TOD0372.ADA	TOD0373.ADA
TOD0380.ADA	TOD0381.ADA	TOD0382.ADA	TOD0383.ADA	TOD0523.ADA	TOD0524.ADA
TOD0525.ADA	TOD0526.ADA	TOD0527.ADA	TOD0528.ADA	TOD0529.ADA	MATHFUN.DEC
ALD9204.DEC	ALD9214.DEC	ALD9224.DEC	ALD9234.DEC	ALDE2C3.DEC	ALDE2C4.DEC
ALDE2C7.DEC	ALDE2C8.DEC	ALDE4C3.DEC	ALDE4C4.DEC	ALDE4C7.DEC	ALDE4C8.DEC
CLD9204.DEC	CLD9214.DEC	CLD9224.DEC	CLD9234.DEC	CLDE2C3.DEC	CLDE2C4.DEC
CLDE2C7.DEC	CLDE2C8.DEC	CLDE4C3.DEC	CLDE4C4.DEC	CLDE4C7.DEC	CLDE4C8.DEC
ELD9204.DEC	ELD9214.DEC	ELD9224.DEC	ELD9234.DEC	ELDE2C3.DEC	ELDE2C4.DEC
ELDE2C7.DEC	ELDE2C8.DEC	ELDE4C3.DEC	ELDE4C4.DEC	ELDE4C7.DEC	ELDE4C8.DEC
SLD9204.DEC	SLD9214.DEC	SLD9224.DEC	SLD9234.DEC	SLDE2C3.DEC	SLDE2C4.DEC
SLDE2C7.DEC	SLDE2C8.DEC	SLDE4C3.DEC	SLDE4C4.DEC	SLDE4C7.DEC	SLDE4C8.DEC
TLD9204.DEC	TLD9214.DEC	TLD9224.DEC	TLD9234.DEC	TLDE2C3.DEC	TLDE2C4.DEC
TLDE2C7.DEC	TLDE2C8.DEC	TLDE4C3.DEC	TLDE4C4.DEC	TLDE4C7.DEC	TLDE4C8.DEC
AGD0059.DEC	CGD0059.DEC	EGD0059.DEC	SGD0059.DEC	TGD0059.DEC	AFD9C03.DEC
AFD9C04.DEC	CFD9C03.DEC	CFD9C04.DEC	efd9C03.DEC	efd9C04.DEC	SFD9C03.DEC
SFD9C04.DEC	TFD9C03.DEC	TFD9C04.DEC	OURSPC.SPA	OURSYS.SPA	OURSYSR.SPA
OURTYP.SPA	OURTYPR.SPA	OURSPC.TIM	OURSYS.TIM	OURSYSR.TIM	OURTYP.TIM
OURTYPR.TIM	ADASYS.USE	ADATYP.USE	AFD4128.USE	AFD451U.USE	AFD4803.USE
AFDC107.USE	AFDD608.USE	CFD4128.USE	CFD451U.USE	CFD4803.USE	CFDC107.USE
CFDD608.USE	efd4128.USE	efd451U.USE	efd4128.USE	efd451U.USE	OURSPC.USE
OURTYP.USE	OURTYPR.USE	SFD4128.USE	SFD451U.USE	SFD4803.USE	SFDC107.USE
SFDD608.USE	TFD4128.USE	TFD451U.USE	TFD4803.USE	TFDC107.USE	TFDD608.USE
FA0P000.CMN	FF0P000.CMN	FFDE000.CMN	FG0P000.CMN	FGD0000.CMN	F000002.CMN
F00P000.CMN	OURFOR.CMN	SYSFOR.CMN	FA00000.FOR	FF03500.FOR	FFC3502.FOR
FF03503.FOR	FF03504.FOR	FFC3514.FOR	FF03519.FOR	FF03600.FOR	FF03602.FOR
FF03603.FOR	FF03605.FOR	FF03606.FOR	FF03608.FOR	FF03609.FOR	FF03620.FOR
FF04120.FOR	FF04121.FOR	FF04122.FOR	FF04123.FOR	FF04124.FOR	FF04125.FOR
FF04126.FOR	FF04127.FOR	FF04129.FOR	FF04510.FOR	FF04511.FOR	FF04512.FOR
FF04513.FOR	FF04514.FOR	FF04515.FOR	FF04516.FOR	FF04517.FOR	FF04518.FOR
FF04519.FOR	FF0451A.FOR	FF0451B.FOR	FF0451C.FOR	FF0451D.FOR	FF0451E.FOR
FF0451F.FOR	FF0451G.FOR	FF0451H.FOR	FF0451I.FOR	FF0451J.FOR	FF0451V.FOR
FF0451W.FOR	FF0451X.FOR	FF0451Y.FOR	FF0451Z.FOR	FF04520.FOR	FF04521.FOR
FF04522.FOR	FF04529.FOR	FF04530.FOR	FF04531.FOR	FF04532.FOR	FF04533.FOR
FF04534.FOR	FF04535.FOR	FF04536.FOR	FF04537.FOR	FF04538.FOR	FF04539.FOR
FF0453A.FOR	FF04540.FOR	FF04541.FOR	FF04550.FOR	FF04551.FOR	FF04552.FOR
FF04553.FOR	FF04554.FOR	FF04555.FOR	FF04556.FOR	FF04557.FOR	FF04558.FOR
FF04559.FOR	FF04560.FOR	FF04562.FOR	FF04563.FOR	FF04600.FOR	FF04601.FOR
FF04602.FOR	FF04601.FOR	FF05200.FOR	FF05201.FOR	FF05202.FOR	FF05203.FOR

Figure A-4. ACPS ANSI Tape Contents Volume ACPS04 (Concluded)

FF05204.FOR	FF05205.FOR	FF05206.FOR	FF05207.FOR	FF05208.FOR	FF05209.FOR
FF0520A.FOR	FF05210.FOR	FF05211.FOR	FF05212.FOR	FF05213.FOR	FF05214.FOR
FF05215.FOR	FF0521H.FOR	FF0521I.FOR	FF0521J.FOR	FF0521K.FOR	FF05300.FOR
FF05301.FOR	FF05302.FOR	FF05303.FOR	FF05304.FOR	FF05305.FOR	FF05306.FOR
FF05307.FOR	FF05308.FOR	FF05400.FOR	FF05401.FOR	FF05402.FOR	FF05403.FOR
FF05404.FOR	FF05405.FOR	FF05406.FOR	FF05407.FOR	FF05408.FOR	FF05501.FOR
FF05502.FOR	FF05503.FOR	FF05504.FOR	FF05505.FOR	FF05506.FOR	FF05507.FOR
FF05508.FOR	FF05509.FOR	FF0550A.FOR	FF0550B.FOR	FF0550C.FOR	FF0550D.FOR
FF0550E.FOR	FF0550F.FOR	FF06001.FOR	FF06009.FOR	FF06010.FOR	FF06011.FOR
FF06013.FOR	FF06014.FOR	FF06015.FOR	FF06016.FOR	FF06017.FOR	FF06018.FOR
FF06019.FOR	FF06022.FOR	FF06023.FOR	FF06024.FOR	FF06025.FOR	FF06026.FOR
FF06027.FOR	FF06028.FOR	FF06029.FOR	FF06030.FOR	FF06031.FOR	FF06032.FOR
FF06033.FOR	FF06043.FOR	FF06044.FOR	FF06045.FOR	FF06046.FOR	FF06047.FOR
FF06048.FOR	FF06049.FOR	FF06050.FOR	FF06051.FOR	FF06052.FOR	FF06053.FOR
FF06060.FOR	FF06061.FOR	FF06062.FOR	FF06063.FOR	FF06064.FOR	FF06065.FOR
FF06066.FOR	FF06067.FOR	FF06068.FOR	FF06069.FOR	FF06070.FOR	FF06071.FOR
FF06072.FOR	FF06073.FOR	FF06074.FOR	FF06075.FOR	FF06076.FOR	FF06077.FOR
FF06078.FOR	FF06079.FOR	FF06100.FOR	FF06101.FOR	FF06108.FOR	FF06109.FOR
FF06110.FOR	FF06111.FOR	FF06112.FOR	FF06113.FOR	FF06114.FOR	FF06115.FOR
FF06116.FOR	FF06117.FOR	FF06118.FOR	FF06119.FOR	FF06122.FOR	FF06132.FOR
FF06133.FOR	FF06134.FOR	FF06135.FOR	FF06136.FOR	FF06137.FOR	FF06138.FOR
FF06139.FOR	FF06140.FOR	FF06142.FOR	FF06143.FOR	FF06144.FOR	FF06145.FOR
FF06146.FOR	FF06147.FOR	FF06148.FOR	FF06149.FOR	FF06150.FOR	FF06152.FOR
FF06153.FOR	FF06154.FOR	FF06155.FOR	FF06156.FOR	FF06157.FOR	FF06158.FOR
FF06159.FOR	FF06160.FOR	FF06162.FOR	FF06163.FOR	FF06164.FOR	FF06165.FOR

Total of 1404 files.

Table A-5. ACPS ANSI Tape Contents Volume APCSOS

FF06166.FOR	FF06167.FOR	FF06168.FOR	FF06169.FOR	FF06170.FOR	FF06172.FOR
FF06173.FOR	FF06174.FOR	FF06175.FOR	FF06176.FOR	FF06177.FOR	FF06178.FOR
FF06179.FOR	FF06180.FOR	FF06182.FOR	FF06183.FOR	FF06184.FOR	FF06185.FOR
FF06186.FOR	FF06187.FOR	FF06188.FOR	FF06189.FOR	FF06190.FOR	FF06802.FOR
FF06803.FOR	FF06806.FOR	FF06807.FOR	FF06816.FOR	FF06817.FOR	FF06818.FOR
FF06819.FOR	FF06820.FOR	FF06821.FOR	FF0D720.FOR	FF0D721.FOR	FF0D727.FOR
FF0D728.FOR	FFD451K.FOR	FFD451L.FOR	FFD451M.FOR	FFD451N.FOR	FFD451O.FOR
FFD451P.FOR	FFD451Q.FOR	FFD451R.FOR	FFD451S.FOR	FFD4523.FOR	FFD4524.FOR
FFD4525.FOR	FFD4526.FOR	FFD455D.FOR	FFD455E.FOR	FFD455F.FOR	FFD455G.FOR
FFD455I.FOR	FFD455J.FOR	FFD455K.FOR	FFD455L.FOR	FFD4567.FOR	FFD4568.FOR
FFD4569.FOR	FFD4603.FOR	FFD4604.FOR	FFD4605.FOR	FFD4606.FOR	FFD4607.FOR
FFD4608.FOR	FFD4609.FOR	FFD460A.FOR	FFD460B.FOR	FFD520C.FOR	FFD520D.FOR
FFD520E.FOR	FFD520F.FOR	FFD520G.FOR	FFD520H.FOR	FFD520I.FOR	FFD520J.FOR
FFD5218.FOR	FFD5219.FOR	FFD521A.FOR	FFD521B.FOR	FFD521C.FOR	FFD521D.FOR
FFD521E.FOR	FFD521F.FOR	FFDD722.FOR	FFDD723.FOR	FFDD724.FOR	FFDD725.FOR
FFDE201.FOR	FFDE202.FOR	FFDE203.FOR	FFDE204.FOR	FFDE205.FOR	FFDE206.FOR
FFDE207.FOR	FFDE208.FOR	FFDE209.FOR	FFDE20A.FOR	FFDE211.FOR	FFDE212.FOR
FFDE213.FOR	FFDE214.FOR	FFDE215.FOR	FFDE216.FOR	FFDE220.FOR	FFDE231.FOR
FFDE232.FOR	FFDE233.FOR	FFDE234.FOR	FFDE235.FOR	FFDE236.FOR	FFDE237.FOR
FFDE238.FOR	FFDE239.FOR	FFDE23A.FOR	FFDE241.FOR	FFDE242.FOR	FFDE243.FOR
FFDE244.FOR	FFDE245.FOR	FFDE246.FOR	FFDE250.FOR	FFDE401.FOR	FFDE402.FOR
FFDE403.FOR	FFDE404.FOR	FFDE405.FOR	FFDE406.FOR	FFDE407.FOR	FFDE408.FOR
FFDE409.FOR	FFDE40A.FOR	FFDE411.FOR	FFDE412.FOR	FFDE413.FOR	FFDE414.FOR
FFDE415.FOR	FFDE416.FOR	FFDE420.FOR	FFDE431.FOR	FFDE432.FOR	FFDE433.FOR
FFDE434.FOR	FFDE435.FOR	FFDE436.FOR	FFDE437.FOR	FFDE438.FOR	FFDE439.FOR
FFDE43A.FOR	FFDE441.FOR	FFDE442.FOR	FFDE443.FOR	FFDE444.FOR	FFDE445.FOR
FFDE446.FOR	FFDE450.FOR	FFDF000.FOR	FFDF001.FOR	FFDF002.FOR	FFDF003.FOR
FFDF004.FOR	FFDF005.FOR	FFIRST.FOR	FGD0001.FOR	FGD000D.FOR	FGD000E.FOR
F000102.FOR	F000103.FOR	F000104.FOR	F000202.FOR	F000203.FOR	F000204.FOR
F000300.FOR	F000305.FOR	F000310.FOR	F000311.FOR	F000312.FOR	F000313.FOR
F000314.FOR	F000315.FOR	F000316.FOR	F000317.FOR	F000318.FOR	F000319.FOR
F000320.FOR	F000321.FOR	F000322.FOR	F000323.FOR	F000324.FOR	F000325.FOR
F000326.FOR	F000327.FOR	F000328.FOR	F000329.FOR	F000330.FOR	F000331.FOR
F000332.FOR	F000333.FOR	F000334.FOR	F000335.FOR	F000336.FOR	F000337.FOR
F000338.FOR	F000339.FOR	F000340.FOR	F000341.FOR	F000342.FOR	F000343.FOR
F000347.FOR	F000348.FOR	F000349.FOR	F000350.FOR	F000351.FOR	F000352.FOR
F000353.FOR	F000354.FOR	F000355.FOR	F000356.FOR	F000357.FOR	F000358.FOR
F000359.FOR	F000360.FOR	F000361.FOR	F000362.FOR	F000390.FOR	F000391.FOR
F000402.FOR	F000403.FOR	F000404.FOR	F000500.FOR	F000502.FOR	F000503.FOR
F000504.FOR	F000507.FOR	F000508.FOR	F000509.FOR	F000512.FOR	F000513.FOR
F000514.FOR	F000517.FOR	F000518.FOR	F000519.FOR	F000602.FOR	F000603.FOR
F000604.FOR	F000605.FOR	F000606.FOR	F000607.FOR	F000702.FOR	F000703.FOR
F000704.FOR	F000705.FOR	F000706.FOR	F000707.FOR	F000708.FOR	F000709.FOR
F000710.FOR	F000711.FOR	F000713.FOR	F000714.FOR	F000715.FOR	F000716.FOR
F000717.FOR	F000718.FOR	F000719.FOR	F000720.FOR	F000721.FOR	F0D0344.FOR
F0D0345.FOR	F0D0346.FOR	F0D036X.FOR	F0D036Y.FOR	F0D036Z.FOR	FORDMP.FOR
FORSYS.FOR	SYSBLK.FOR	FFD451U.USE	FFD4566.USE	FORTYP.USE	TYPFOR.USE
JAOP000.CPL	JFOP000.CPL	JFJ3500.CPL	JFJ3501.CPL	JFJ3502.CPL	JFJ3503.CPL
JFJ3504.CPL	JFJ3513.CPL	JFJ3514.CPL	JFJ3600.CPL	JFJ3601.CPL	JFJ3602.CPL
JFJ3603.CPL	JFJ3604.CPL	JFJ3605.CPL	JFJ3606.CPL	JFJ3607.CPL	JFJ3608.CPL
JFJ3609.CPL	JFJ3610.CPL	JFJ3700.CPL	JFJ3701.CPL	JFJ3702.CPL	JFJ3703.CPL
JFJ3800.CPL	JFJ3801.CPL	JFJ3802.CPL	JFJ3803.CPL	JFJ3804.CPL	JFJ4518.CPL
JFJ4519.CPL	JFJ451A.CPL	JFJ451B.CPL	JFJ451C.CPL	JFJ451D.CPL	JFJ451E.CPL
JFJ451F.CPL	JFJ451G.CPL	JFJ451K.CPL	JFJ451L.CPL	JFJ451M.CPL	JFJ451P.CPL
JFJ451Q.CPL	JFJ451V.CPL	JFJ451W.CPL	JFJ451X.CPL	JFJ451Y.CPL	JFJ5300.CPL
JFJ5301.CPL	JFJ5302.CPL	JFJ5303.CPL	JFJ5305.CPL	JFJ5306.CPL	JFJ5307.CPL
JFJ5400.CPL	JFJ5401.CPL	JFJ5402.CPL	JFJ5403.CPL	JFJ5404.CPL	JFJ5405.CPL
JFJ5406.CPL	JFJ5407.CPL	JFJ5501.CPL	JFJ5502.CPL	JFJ5503.CPL	JFJ5504.CPL
JFJ5506.CPL	JFJ5507.CPL	JFJ5508.CPL	JFJ5509.CPL	JFJ550A.CPL	JFJ550B.CPL
JFJ6013.CPL	JFJ6014.CPL	JFJ6015.CPL	JFJ6016.CPL	JFJ6017.CPL	JFJ6018.CPL
JFJ6019.CPL	JFJ6023.CPL	JFJ6024.CPL	JFJ6025.CPL	JFJ6026.CPL	JFJ6027.CPL
JFJ6028.CPL	JFJ6029.CPL	JFJ6030.CPL	JFJ6031.CPL	JFJ6032.CPL	JFJ6033.CPL
JFJ6044.CPL	JFJ6045.CPL	JFJ6046.CPL	JFJ6047.CPL	JFJ6048.CPL	JFJ6049.CPL
JFJ6050.CPL	JFJ6051.CPL	JFJ6052.CPL	JFJ6060.CPL	JFJ6061.CPL	JFJ6062.CPL
JFJ6063.CPL	JFJ6064.CPL	JFJ6065.CPL	JFJ6066.CPL	JFJ6067.CPL	JFJ6068.CPL
JFJ6070.CPL	JFJ6071.CPL	JFJ6072.CPL	JFJ6073.CPL	JFJ6074.CPL	JFJ6075.CPL
JFJ6076.CPL	JFJ6077.CPL	JFJ6078.CPL	JFJ6100.CPL	JFJ6108.CPL	JFJ6112.CPL
JFJ6113.CPL	JFJ6114.CPL	JFJ6115.CPL	JFJ6116.CPL	JFJ6117.CPL	JFJ6118.CPL
JFJ6119.CPL	JFJ6132.CPL	JFJ6133.CPL	JFJ6134.CPL	JFJ6135.CPL	JFJ6136.CPL

Table A-5. ACPS ANSI Tape Contents Volume APCSOS (Continued)

JFJ6137.CPL	JFJ6138.CPL	JFJ6139.CPL	JFJ6142.CPL	JFJ6143.CPL	JFJ6144.CPL
JFJ6145.CPL	JFJ6146.CPL	JFJ6147.CPL	JFJ6148.CPL	JFJ6149.CPL	JFJ6152.CPL
JFJ6153.CPL	JFJ6154.CPL	JFJ6155.CPL	JFJ6156.CPL	JFJ6157.CPL	JFJ6158.CPL
JFJ6159.CPL	JFJ6162.CPL	JFJ6163.CPL	JFJ6164.CPL	JFJ6165.CPL	JFJ6166.CPL
JFJ6167.CPL	JFJ6168.CPL	JFJ6169.CPL	JFJ6172.CPL	JFJ6173.CPL	JFJ6174.CPL
JFJ6175.CPL	JFJ6176.CPL	JFJ6177.CPL	JFJ6178.CPL	JFJ6179.CPL	JFJ6182.CPL
JFJ6183.CPL	JFJ6184.CPL	JFJ6185.CPL	JFJ6186.CPL	JFJ6187.CPL	JFJ6188.CPL
JFJ6189.CPL	JFJ6191.CPL	JFJ6192.CPL	JFJ6193.CPL	JFJ6194.CPL	JFJ6195.CPL
JFJ6196.CPL	JFJ6197.CPL	JFJ6198.CPL	JFJ619A.CPL	JFJ619B.CPL	JFJ619C.CPL
JFJ619D.CPL	JFJ619E.CPL	JFJ619F.CPL	JFJ619G.CPL	JFJ619H.CPL	JFJ619J.CPL
JFJ619K.CPL	JFJ619L.CPL	JFJ619M.CPL	JFJ619N.CPL	JFJ619O.CPL	JFJ619P.CPL
JFJ619Q.CPL	JFJ6213.CPL	JFJ6214.CPL	JFJ6215.CPL	JFJ6216.CPL	JFJ6217.CPL
JFJ6218.CPL	JFJ6219.CPL	JFJ6223.CPL	JFJ6224.CPL	JFJ6225.CPL	JFJ6226.CPL
JFJ6227.CPL	JFJ6228.CPL	JFJ6229.CPL	JFJ6230.CPL	JFJ6231.CPL	JFJ6232.CPL
JFJ6243.CPL	JFJ6244.CPL	JFJ6245.CPL	JFJ6246.CPL	JFJ6247.CPL	JFJ6248.CPL
JFJ6249.CPL	JFJ6250.CPL	JFJ6251.CPL	JFJ6252.CPL	JFJ6802.CPL	JFJ6806.CPL
JFJ6808.CPL	JFJ680A.CPL	JFJ680C.CPL	JFJ680E.CPL	JFJ680G.CPL	JFJ680I.CPL
JFJ6810.CPL	JFJ6814.CPL	JFJ6816.CPL	JFJ6818.CPL	JFJ6820.CPL	JFJ6822.CPL
JFJ6824.CPL	JG0P000.CPL	JGD0000.CPL	JGD0004.CPL	JGD0005.CPL	JGD0006.CPL
JGJ000D.CPL	JGN000F.CPL	JGN000G.CPL	JGN000H.CPL	J000002.CPL	J00P000.CPL
JOJ0102.CPL	JOJ0103.CPL	JOJ0202.CPL	JOJ0203.CPL	JOJ0311.CPL	JOJ0312.CPL
JOJ0314.CPL	JOJ0315.CPL	JOJ0323.CPL	JOJ0324.CPL	JOJ0326.CPL	JOJ0327.CPL
JOJ0329.CPL	JOJ0330.CPL	JOJ0332.CPL	JOJ0333.CPL	JOJ0335.CPL	JOJ0336.CPL
JOJ0338.CPL	JOJ0339.CPL	JOJ0341.CPL	JOJ0342.CPL	JOJ0344.CPL	JOJ0345.CPL
JOJ0347.CPL	JOJ0348.CPL	JOJ0349.CPL	JOJ0351.CPL	JOJ0352.CPL	JOJ0363.CPL
JOJ0364.CPL	JOJ036I.CPL	JOJ036J.CPL	JOJ036L.CPL	JOJ036M.CPL	JOJ036O.CPL
JOJ036P.CPL	JOJ036R.CPL	JOJ036S.CPL	JOJ036U.CPL	JOJ036V.CPL	JOJ036X.CPL
JOJ036Y.CPL	JOJ0390.CPL	JOJ0402.CPL	JOJ0403.CPL	JOJ0502.CPL	JOJ0503.CPL
JOJ0507.CPL	JOJ0508.CPL	JOJ0512.CPL	JOJ0513.CPL	JOJ0517.CPL	JOJ0518.CPL
JOJ0523.CPL	JOJ0524.CPL	JOJ0602.CPL	JOJ0603.CPL	JOJ0702.CPL	JOJ0703.CPL
JOJ0706.CPL	JOJ0707.CPL	JOJ0709.CPL	JOJ0710.CPL	JOJ0719.CPL	JOJ0720.CPL
JOVDMP.CPL	JOVSPC.CPL	OURJOV.CPL	DMPJOV.CPL	JA00000.JOV	JF03500.JOV
JF03501.JOV	JF03502.JOV	JF03503.JOV	JF03504.JOV	JF03513.JOV	JF03514.JOV
JF03519.JOV	JF03550.JOV	JF03600.JOV	JF03601.JOV	JF03602.JOV	JF03603.JOV
JF03604.JOV	JF03605.JOV	JF03606.JOV	JF03607.JOV	JF03608.JOV	JF03609.JOV
JF03610.JOV	JF03620.JOV	JF03700.JOV	JF03701.JOV	JF03702.JOV	JF03703.JOV
JF03704.JOV	JF03800.JOV	JF03801.JOV	JF03802.JOV	JF03803.JOV	JF03804.JOV
JF03805.JOV	JF04120.JOV	JF04121.JOV	JF04122.JOV	JF04123.JOV	JF04124.JOV
JF04125.JOV	JF04126.JOV	JF04127.JOV	JF04129.JOV	JF0412B.JOV	JF04130.JOV
JF04131.JOV	JF04132.JOV	JF04133.JOV	JF0413D.JOV	JF0413H.JOV	JF0413I.JOV
JF0413J.JOV	JF0413K.JOV	JF04310.JOV	JF04311.JOV	JF04312.JOV	JF04510.JOV
JF04511.JOV	JF04512.JOV	JF04513.JOV	JF04514.JOV	JF04515.JOV	JF04516.JOV
JF04517.JOV	JF04518.JOV	JF04519.JOV	JF0451A.JOV	JF0451B.JOV	JF0451C.JOV
JF0451D.JOV	JF0451E.JOV	JF0451F.JOV	JF0451G.JOV	JF0451J.JOV	JF0451V.JOV
JF0451W.JOV	JF0451X.JOV	JF0451Y.JOV	JF0451Z.JOV	JF04520.JOV	JF04521.JOV
JF04522.JOV	JF04527.JOV	JF04529.JOV	JF04530.JOV	JF04531.JOV	JF04532.JOV
JF04533.JOV	JF04534.JOV	JF04535.JOV	JF04536.JOV	JF04537.JOV	JF04538.JOV
JF04539.JOV	JF0453A.JOV	JF0453B.JOV	JF0453C.JOV	JF04540.JOV	JF04541.JOV
JF04550.JOV	JF04551.JOV	JF04552.JOV	JF04553.JOV	JF04554.JOV	JF04555.JOV
JF04556.JOV	JF04557.JOV	JF04558.JOV	JF04559.JOV	JF0455N.JOV	JF0455O.JOV
JF04560.JOV	JF04562.JOV	JF04563.JOV	JF04600.JOV	JF04601.JOV	JF04602.JOV
JF0460E.JOV	JF0460F.JOV	JF0460G.JOV	JF0460H.JOV	JF0460I.JOV	JF0460J.JOV
JF0460M.JOV	JF05200.JOV	JF05201.JOV	JF05202.JOV	JF05203.JOV	JF05204.JOV
JF05205.JOV	JF05206.JOV	JF05207.JOV	JF05208.JOV	JF05209.JOV	JF0520A.JOV
JF0520B.JOV	JF0520M.JOV	JF05200.JOV	JF0520R.JOV	JF0520S.JOV	JF05210.JOV
JF05211.JOV	JF05212.JOV	JF05213.JOV	JF05214.JOV	JF05215.JOV	JF0521H.JOV
JF0521I.JOV	JF0521J.JOV	JF0521K.JOV	JF0521L.JOV	JF0521N.JOV	JF05300.JOV
JF05301.JOV	JF05302.JOV	JF05303.JOV	JF05304.JOV	JF05305.JOV	JF05306.JOV
JF05307.JOV	JF05308.JOV	JF05400.JOV	JF05401.JOV	JF05402.JOV	JF05403.JOV
JF05404.JOV	JF05405.JOV	JF05406.JOV	JF05407.JOV	JF05408.JOV	JF05501.JOV
JF05502.JOV	JF05503.JOV	JF05504.JOV	JF05505.JOV	JF05506.JOV	JF05507.JOV
JF05508.JOV	JF05509.JOV	JF0550A.JOV	JF0550B.JOV	JF0550C.JOV	JF0550D.JOV
JF0550E.JOV	JF0550F.JOV	JF06001.JOV	JF06009.JOV	JF06010.JOV	JF06011.JOV
JF06013.JOV	JF06014.JOV	JF06015.JOV	JF06016.JOV	JF06017.JOV	JF06018.JOV
JF06019.JOV	JF06022.JOV	JF06023.JOV	JF06024.JOV	JF06025.JOV	JF06026.JOV
JF06027.JOV	JF06028.JOV	JF06029.JOV	JF06030.JOV	JF06031.JOV	JF06032.JOV
JF06033.JOV	JF06043.JOV	JF06044.JOV	JF06045.JOV	JF06046.JOV	JF06047.JOV
JF06048.JOV	JF06049.JOV	JF06050.JOV	JF06051.JOV	JF06052.JOV	JF06053.JOV
JF06060.JOV	JF06061.JOV	JF06062.JOV	JF06063.JOV	JF06064.JOV	JF06065.JOV
JF06066.JOV	JF06067.JOV	JF06068.JOV	JF06069.JOV	JF06070.JOV	JF06071.JOV

Table A-5. ACPS ANSI Tape Contents Volume APCSOS (Continued)

JF06072.JOV	JF06073.JOV	JF06074.JOV	JF06075.JOV	JF06076.JOV	JF06077.JOV
JF06078.JOV	JF06079.JOV	JF06100.JOV	JF06101.JOV	JF06108.JOV	JF06109.JOV
JF06110.JOV	JF06111.JOV	JF06112.JOV	JF06113.JOV	JF06114.JOV	JF06115.JOV
JF06116.JOV	JF06117.JOV	JF06118.JOV	JF06119.JOV	JF06122.JOV	JF06132.JOV
JF06133.JOV	JF06134.JOV	JF06135.JOV	JF06136.JOV	JF06137.JOV	JF06138.JOV
JF06139.JOV	JF06140.JOV	JF06142.JOV	JF06143.JOV	JF06144.JOV	JF06145.JOV
JF06146.JOV	JF06147.JOV	JF06148.JOV	JF06149.JOV	JF06150.JOV	JF06152.JOV
JF06153.JOV	JF06154.JOV	JF06155.JOV	JF06156.JOV	JF06157.JOV	JF06158.JOV
JF06159.JOV	JF06160.JOV	JF06162.JOV	JF06163.JOV	JF06164.JOV	JF06165.JOV
JF06166.JOV	JF06167.JOV	JF06168.JOV	JF06169.JOV	JF06170.JOV	JF06172.JOV
JF06173.JOV	JF06174.JOV	JF06175.JOV	JF06176.JOV	JF06177.JOV	JF06178.JOV
JF06179.JOV	JF06180.JOV	JF06182.JOV	JF06183.JOV	JF06184.JOV	JF06185.JOV
JF06186.JOV	JF06187.JOV	JF06188.JOV	JF06189.JOV	JF06190.JOV	JF06191.JOV
JF06192.JOV	JF06193.JOV	JF06194.JOV	JF06195.JOV	JF06196.JOV	JF06197.JOV
JF06198.JOV	JF06199.JOV	JF0619A.JOV	JF0619B.JOV	JF0619C.JOV	JF0619D.JOV
JF0619E.JOV	JF0619F.JOV	JF0619G.JOV	JF0619H.JOV	JF0619I.JOV	JF0619J.JOV
JF0619K.JOV	JF0619L.JOV	JF0619M.JOV	JF0619N.JOV	JF0619O.JOV	JF0619P.JOV
JF0619Q.JOV	JF0619R.JOV	JF06802.JOV	JF06803.JOV	JF06806.JOV	JF06807.JOV
JF06808.JOV	JF06809.JOV	JF0680A.JOV	JF0680B.JOV	JF0680C.JOV	JF0680D.JOV
JF0680E.JOV	JF0680F.JOV	JF06810.JOV	JF06811.JOV	JF06814.JOV	JF06815.JOV
JF06816.JOV	JF06817.JOV	JF06818.JOV	JF06819.JOV	JF06820.JOV	JF06821.JOV
JF06822.JOV	JF06823.JOV	JF06824.JOV	JF06825.JOV	JF0D720.JOV	JF0D721.JOV
JF0D727.JOV	JF0D728.JOV	JF0D729.JOV	JF0D72D.JOV	JFD3551.JOV	JFD4128.JOV
JFD412A.JOV	JFD412C.JOV	JFD412D.JOV	JFD4135.JOV	JFD4136.JOV	JFD4137.JOV
JFD4138.JOV	JFD4139.JOV	JFD413A.JOV	JFD413B.JOV	JFD413C.JOV	JFD413E.JOV
JFD413G.JOV	JFD451K.JOV	JFD451L.JOV	JFD451M.JOV	JFD451P.JOV	JFD451Q.JOV
JFD4523.JOV	JFD4524.JOV	JFD452A.JOV	JFD455D.JOV	JFD455E.JOV	JFD455I.JOV
JFD455J.JOV	JFD4566.JOV	JFD4567.JOV	JFD4603.JOV	JFD4604.JOV	JFD4605.JOV
JFD4606.JOV	JFD4607.JOV	JFD460K.JOV	JFD460L.JOV	JFD460N.JOV	JFD460O.JOV
JFD520C.JOV	JFD520D.JOV	JFD520E.JOV	JFD520F.JOV	JFD520N.JOV	JFD520P.JOV
JFD520Q.JOV	JFD5218.JOV	JFD5219.JOV	JFD521A.JOV	JFD521B.JOV	JFD521K.JOV
JFD521M.JOV	JFD521O.JOV	JFD521P.JOV	JFD6201.JOV	JFD6209.JOV	JFD6210.JOV
JFD6211.JOV	JFD6213.JOV	JFD6214.JOV	JFD6215.JOV	JFD6216.JOV	JFD6217.JOV
JFD6218.JOV	JFD6219.JOV	JFD6222.JOV	JFD6223.JOV	JFD6224.JOV	JFD6225.JOV
JFD6226.JOV	JFD6227.JOV	JFD6228.JOV	JFD6229.JOV	JFD6230.JOV	JFD6231.JOV
JFD6232.JOV	JFD6233.JOV	JFD6243.JOV	JFD6244.JOV	JFD6245.JOV	JFD6246.JOV
JFD6247.JOV	JFD6248.JOV	JFD6249.JOV	JFD6250.JOV	JFD6251.JOV	JFD6252.JOV
JFD6253.JOV	JFD680G.JOV	JFD680H.JOV	JFD680I.JOV	JFD680J.JOV	JFDD600.JOV
JFDD601.JOV	JFDD602.JOV	JFDD603.JOV	JFDD604.JOV	JFDD605.JOV	JFDD606.JOV
JFDD607.JOV	JFDD608.JOV	JFDD609.JOV	JFDD610.JOV	JFDD611.JOV	JFDD722.JOV
JFDD723.JOV	JFDD72A.JOV	JFDD72B.JOV	JFDD72C.JOV	JFDD72E.JOV	JFDD72F.JOV
JFDD72G.JOV	JFDD72H.JOV	JFDD72I.JOV	JFDDA01.JOV	JFDDA02.JOV	JFDF000.JOV
JFDF001.JOV	JFDF002.JOV	JFDF003.JOV	JFDF004.JOV	JFDF005.JOV	JFIRST.JOV
JGD000C.JOV	JGD0001.JOV	JGD0007.JOV	JGD0008.JOV	JGD0009.JOV	JGD000D.JOV
JGD000E.JOV	JGN0001.JOV	JGN000J.JOV	JGN000K.JOV	J000102.JOV	J000103.JOV
J000104.JOV	J000202.JOV	J000203.JOV	J000204.JOV	J000300.JOV	J000305.JOV
J000310.JOV	J000311.JOV	J000312.JOV	J000313.JOV	J000314.JOV	J000315.JOV
J000316.JOV	J000323.JOV	J000324.JOV	J000325.JOV	J000326.JOV	J000327.JOV
J000328.JOV	J000329.JOV	J000330.JOV	J000331.JOV	J000332.JOV	J000333.JOV
J000334.JOV	J000335.JOV	J000336.JOV	J000337.JOV	J000338.JOV	J000339.JOV
J000340.JOV	J000341.JOV	J000342.JOV	J000343.JOV	J000347.JOV	J000348.JOV
J000349.JOV	J000350.JOV	J000351.JOV	J000352.JOV	J000353.JOV	J000363.JOV
J000364.JOV	J000365.JOV	J00036I.JOV	J00036J.JOV	J00036K.JOV	J00036L.JOV
J00036M.JOV	J00036N.JOV	J00036O.JOV	J00036V.JOV	J00036W.JOV	J000390.JOV
J000391.JOV	J000402.JOV	J000403.JOV	J000404.JOV	J000500.JOV	J000502.JOV
J000503.JOV	J000504.JOV	J000507.JOV	J000508.JOV	J000509.JOV	J000512.JOV
J000513.JOV	J000514.JOV	J000517.JOV	J000518.JOV	J000519.JOV	J000602.JOV
J000603.JOV	J000604.JOV	J000605.JOV	J000606.JOV	J000607.JOV	J000702.JOV
J000703.JOV	J000704.JOV	J000705.JOV	J000706.JOV	J000707.JOV	J000708.JOV
J000709.JOV	J000710.JOV	J000711.JOV	J000713.JOV	J000714.JOV	J000715.JOV
J000716.JOV	J000717.JOV	J000718.JOV	J000719.JOV	J000720.JOV	J000721.JOV
J0D0344.JOV	J0D0345.JOV	J0D0346.JOV	J0D0360.JOV	J0D036P.JOV	J0D036Q.JOV
J0D036R.JOV	J0D036S.JOV	J0D036T.JOV	J0D036X.JOV	J0D036Y.JOV	J0D036Z.JOV
J0D0523.JOV	J0D0524.JOV	J0D0525.JOV	J0VSYS.JOV	SPCJOV.JOV	JF03701.USE
JF04311.USE	JF0451H.USE	JF0451I.USE	JF0451J.USE	JF0619A.USE	JF06822.USE
JF0451N.USE	JF0451O.USE	JF0451R.USE	JF0451S.USE	JFD451U.USE	JFD4525.USE
JFD4526.USE	JFD4528.USE	JFD455F.USE	JFD455G.USE	JFD455K.USE	JFD455L.USE
JFD4568.USE	JFD4569.USE	JFD4608.USE	JFD4609.USE	JFD460A.USE	JFD460B.USE
JFD460C.USE	JFD460D.USE	JFD520G.USE	JFD520H.USE	JFD520I.USE	JFD520J.USE
JFD521C.USE	JFD521D.USE	JFD521E.USE	JFD521F.USE	JFD6260.USE	JFD6261.USE

Table A-5. ACPS ANSI Tape Contents Volume APCSOS (Concluded)

JFD6262.USE	JFD6263.USE	JFD6264.USE	JFD6265.USE	JFD6266.USE	JFD6267.USE
JFD6268.USE	JFD6269.USE	JFD6270.USE	JFD6271.USE	JFD6272.USE	JFD6273.USE
JFD6274.USE	JFD6275.USE	JFD6276.USE	JFD6277.USE	JFD6278.USE	JFD6279.USE
JFDD602.USE	JFDD603.USE	JFDD610.USE	JFDD611.USE	JFDD724.USE	JFDD725.USE
JFJ451H.USE	JFJ451I.USE	JFJ451N.USE	JFJ451O.USE	JFJ451R.USE	JFJ451S.USE
JFJ6260.USE	JFJ6261.USE	JFJ6262.USE	JFJ6263.USE	JFJ6264.USE	JFJ6265.USE
JFJ6266.USE	JFJ6267.USE	JFJ6268.USE	JFJ6270.USE	JFJ6271.USE	JFJ6272.USE
JFJ6273.USE	JFJ6274.USE	JFJ6275.USE	JFJ6276.USE	JFJ6277.USE	JFJ6278.USE
JGD0008.USE	JGN000J.USE	J000317.USE	J000318.USE	J000319.USE	J000320.USE
J000321.USE	J000322.USE	J0J0317.USE	J0J0318.USE	J0J0320.USE	J0J0321.USE
JOVTYP.USE	TYPJOV.USE	JGF0001.FOR	JGF000E.FOR	SYSJOV.FOR	TMPDMP.FOR
JOVMATH.DAT	CCOMPA.CMN	CCOMPV.CMN	CEXECA.CMN	CEXECV.CMN	CCOMP.FOR
CEXEC.FOR					

Total of 1327 files.

APPENDIX B

DEC VAX/VMS BACKUP TAPE FORMAT

This appendix describes the delivery tape format and directory file organization for the DEC VAX/VMS version of the ACPS.

One tape is written in DEC VAX/VMS BACKUP format as volume ACPS, save set ACPS.BCK. Refer to the DEC VAX/VMS Backup Utility Reference Manual for details on BACKUP formatted tapes. The tape is structured as a multi-file, multi-directory, single-volume configuration. The different files represent collections of ACPS test programs and command procedures. Each collection of similar files is grouped together in specific directories on the tape. The organization of the files on this tape according to directory is as follows:

[acps]	Top-level ACPS directory containing command procedures to establish a common language test environment and to display the VAX/VMS software environment.
[acps.results]	Contains output files from ACPS compilations and test executions and from ACPS test comparison tool executions.
[acps.support]	Contains common language test support software and command procedures.
[acps.tools]	Contains ACPS test comparison tool program source and command procedures files.
[acps.ada]	Top-level directory for Ada tests.
[acps.ada.dec]	Contains Ada test files and command procedures for use with the DEC VAX Ada compiler.
[acps.ada.dec.lib], [acps.ada.dec.lib.sub]	Reserved for program library files generated by VAX Ada compilations of Ada tests.
[acps.fortran]	Contains FORTRAN test files and command procedures for use with the DEC VAX FORTRAN compiler.
[acps.jovial]	Contains JOVIAL test files and command procedures for use with the ECSPO JOVIAL compiler hosted and targeted to VAX/VMS.
[acps.jovial.com.user]	Contains a command procedure to define logical names and global symbols for execution of the ECSPO JOVIAL compiler as described in Appendix A of the EMAD ITS JOVIAL J73 Compiler User's Guide.

APPENDIX C

VAX/VMS SAMPLE COMMAND PROCEDURES

This appendix describes use of a set of VAX/VMS 4.X command procedures that are provided on the ACPS BACKUP formatted delivery tape to perform the tasks necessary to compile/execute ACPS tests and to use the ACPS test comparison tools. Listings are provided for each command procedure discussed. The test compilation and execution command procedures shown are for DEC VAX Ada and were used to generate the output examples discussed in Appendix E. Table C-1 lists the file names (using ACPS logical name conventions) and purpose of all VAX/VMS command procedures provided with the ACPS.

This appendix assumes familiarity with the VAX/VMS command language (DCL). The information presented should be useful to those desiring to either rehost the ACPS or to use it for an Ada compiler other than DEC VAX Ada.

C.1 INSTALLATION OF ACPS SOURCE AND COMMAND FILES

For VAX/VMS users, the ACPS delivery tape is written in BACKUP utility format and contains the single save set ACPS.BCK.

To install the tape the user should type the following:

```
$MOUNT/FOREIGN MTA0:  
$BACKUP/LOG MTA0:ACPS.BCK MYDISK:[MYDIRECTORY...]*
```

where:

MTA0 is the tape drive on which the ACPS tape is mounted.
MYDISK is a user disk.
MYDIRECTORY is a VMS directory to contain the ACPS.

The directory [MYDIRECTORY.ACPS] will be created as the top-level ACPS directory. The following global symbol definition must be placed in the user's LOGIN.COM file:

```
$acps==@MYDISK:[MYDIRECTORY.ACPS]LOGICALS MYDISK MYDIRECTORY
```

All ACPS batch command procedures use the symbol ACPS to establish an initial test environment state common to all ACPS test languages. Its use causes invocation of the ACPS command procedure LOGICALS.COM which defines all ACPS logical names and test environment global symbols, and sets the default directory to the top-level ACPS directory. Table C-2 contains a listing of LOGICALS.COM. Subsequent sections in this appendix will refer to the logical names defined in this command procedure.

C.2 INSTALLATION OF COMMON LANGUAGE SUPPORT SOFTWARE

A separate directory in ACPS is reserved for common language test support software which is used to gather VMS run-time performance statistics and to access microsecond level time. To install this software the user should type the following:

```
$ ACPS
$ SET DEF DSUP
$ @SUPPORT
```

Table C-3 contains a listing of DSUP:SUPPORT.COM.

To execute ACPS microsecond level tests, the MACRO routine contained in file DSUP:USSDISP.MAR must be installed as a user written system service in the VMS directory SYS\$SHARE as file USS.OLB. A template for developing user written system services is contained in file USSDISP.MAR in the VMS directory SYS\$EXAMPLES. If the user is unable to create the necessary system service, then the following must be done in order to execute any of the ACPS tests:

- a. Remove the call to USER_GET_ICR in file DSUP:GETICR.MAR and reexecute DSUP:SUPPORT.
- b. Remove references to SYS\$SHARE:USS.OLB in the following command procedures: DDEC:SETUP.COM, DFOR:SETUP.COM and DJOV:SETUP.COM.

C.3 INSTALLATION OF ACPS TEST COMPARISON TOOLS

To install the ACPS test comparison tools, the following should be typed:

```
$ ACPS
$ SET DEF DTOOL
$ @TOOLS
```

Table C-4 contains a listing of DTOOL:TOOLS.COM.

C.4 DETERMINATION OF VAX/VMS SOFTWARE ENVIRONMENT

In order to draw conclusions about the results of execution of ACPS tests for a given compiler, one must know the VAX/VMS software environment under which the tests are executed. Table C-5 shows a listing of the command procedure DACPS:VAXVMS which is used to display all relevant information about the VMS software environment.

C.5 DETERMINATION OF ADA COMPILER-DEPENDENT PARAMETERS

The Ada run-time command procedures and Ada tests supplied with the ACPS are dependent upon several compiler-specific parameters. The values chosen for these parameters are specific to the DEC VAX Ada compiler. The program that displays these parameters is contained in file DDEC:ADAPARM.ADA and is executed by the command procedure DDEC:ADAPARM.COM which is shown in Table C-6. Table C-7 shows the output from ADAPARM for DEC VAX Ada. The Ada test support packages OURSYS.ADA, ADASYS.ADA, OURTYP.ADA, OURSYSR.ADA and OURTYPR.ADA make the following assumptions:

- a. The resolution of the Ada clock (i.e., the Ada clock cycle or smallest nonzero difference in Ada clock values) is not worse than 10 ms.

In order to minimize test execution times, it is desirable to keep the clock granularity as low as possible. If it is greater than 10 ms, then the variable TCLOCK in package ADASYS.ADA should be initialized to the value "OS".

- b. Run-time statistics are written to the standard output file in Ada which is assumed to be SYS\$OUTPUT.

If this assumption is invalid, then either the approach used in recording run statistics within the ACPS Ada command procedures provided must be changed or all PUT and NEW_LINE statements in the packages ADASYS.ADA and ADADMP.ADA must be changed to output to file SYS\$OUTPUT.

- c. ACPS numeric data types defined in packages OURSYS.ADA, OURSYSR.ADA, OURTYP.ADA, and OURTYPR.ADA are mapped to VAX data types as follows:

<u>ACPS Type</u>	<u>VAX Ada Type</u>	<u>VAX Type-Size</u>
OUR_INTEGER	INTEGER	Longword-32 bits
OUR_FLOAT	FLOAT	F_Float-32 bits
OUR_POSITIVE	POSITIVE	Longword-32 bits
OUR_SHORT_INTEGER	SHORT_INTEGER	Word-16 bits
OUR_SHORT_SHORT_INTEGER	SHORT_SHORT_INTEGER	Byte-8 bits
OUR_LONG_FLOAT	LONG_FLOAT	G_float-64 bits
OUR_LONG_LONG_FLOAT	LONG_LONG_FLOAT	H_float-128 bits
OUR_D_FLOAT	D_FLOAT	D_float-64 bits

If the Ada compiler mapping between predefined ADA numeric types and VAX data types is not as shown in the above table, then the corresponding ACPS type statements in packages OURSYS.ADA, OURSYSR.ADA, OURTYP.ADA, and OURTYPR.ADA must be modified to ensure the correct VAX data type representation for each ACPS numeric type.

- d. The minimum time used in a DELAY statement to force invocation of a lower priority task is assumed to be ten milliseconds. This value establishes the minimum time that will be used in DELAY statements by ACPS tests. Its value is contained in the named constant `minimum_delay_interval` defined in package OURSYS.

C.6 ACPS COMMAND PROCEDURE TECHNIQUES

This section discusses command procedure techniques common to all ACPS test languages and compilers. The discussion refers to the command procedure shown in Table C-6 which compiles and executes the Ada program ADAPARM using DEC VAX Ada compiler.

- a. Creation of compiler-specific test environments. ACPS compilation and execution command procedures are assumed to be executed in batch mode. To create an initial compiler-independent test environment, the first command (shown on line 46 of Fig. C-6) uses the symbol "ACPS" to invoke the command procedure DACPS:LOGICALS.COM which does the following:
 - 1. It defines logical names for each ACPS directory.
 - 2. It defines global symbols (e.g., `USE_DEC`) which are used to establish compiler-specific test environments.
 - 3. It sets the batch output buffer update interval to one day to ensure that VMS batch output buffer flushes do not impact test measurements.
 - 4. It sets the default directory to DACPS.

To create a compiler-specific test environment, the next command (shown on line 47 of Table C-17) uses the symbol `USE_DEC` to invoke the command procedure DDEC:SETUP.COM which completes establishment of the test environment for DEC VAX Ada compiler. As shown in Table C-8, DDEC:SETUP.COM does the following:

- 1. It defines global symbols used by all ACPS compilation and execution command procedures by executing the procedure DSUP:SYMBOLS.COM shown in Table C-9.
- 2. It defines symbols which are used in ACPS compilation command procedures to specify compiler options and common language test support software object modules.
- 3. It sets the current directory to the one containing the command procedures and test cases for a specific compiler.

- b. Creation of compile-time test environment. A common language test environment is created to begin each ACPS compile-time test. All extraneous files are deleted (as shown in lines 38-42 of Table C-6). The working set parameters are set to the specific values shown in line 59 of Table C-6 and are also displayed in the compile-time test statistic file. The system activity before and after the test is also displayed to verify that the compilations are made in a stand-alone environment and are not impacted by other VMS processes.
- c. Creation of run-time test environment. A common language test environment is created for each ACPS run-time test. The working set parameters are set to the specific values shown in line 57 of Table C-6 and are also output to the run-time statistic file. The parameters as shown are used for all single task tests to ensure that VAX automatic working set adjustment logic does not affect test measurement repeatability by interrupting test executions. The system activity before and after the tests is displayed to ensure that test measurements are not affected by other VMS processes or system users.
- d. Recording of run-time test statistics. All ACPS run-time command procedures assume that run-time statistics from test executions are written to the file with logical name SYS\$OUTPUT.

To ensure that an output operation from one test execution does not impact (due to output completion or file extension) another test, a separate command procedure is used to execute each test and to append the results to a single permanent file. As shown in lines 68 and 69 of Table C-6, the logical name RESULTS is assigned to the permanent file that will contain the performance statistics from all test executions. The symbol OUR_RUN (defined by DSUP:SYMBOLS) is used to invoke the command procedure DSUP:RUN which executes a test and appends the results to the file with logical name RESULTS.

Table C-10 contains a listing of DSUP:RUN.COM. As shown at line 32 of Table C-10, DSUP:RUN uses the command procedure DSUP:EXEC, shown in Table C-11, to execute the test program and directs the output to a temporary file. It then uses the APPEND command to append the temporary file to the file RESULTS and finally deletes the temporary file.

To ensure that completion of output from a DCL SHOW command does not impact a test execution, a separate command procedure, DSUP:SHOW, is used to execute each show command and append the output to the file with logical name RESULTS. Table C-12 contains a listing of DSUP:SHOW.COM. As shown at line 26, DSUP:SHOW uses the command procedure SHOW_, shown in Table C-13, to execute the SHOW command and directs the output to a temporary file. The APPEND command is then used to add the temporary file to the end of the file RESULTS. The temporary file is then deleted.

C.7 DETERMINATION OF TEST LOOP OVERHEAD PARAMETERS

The ACPS test comparison tool is used to account for test loop overhead in ACPS tests. For host machines that do not support execution of the tool (i.e., host computers not supporting FORTRAN 77), a special test program in each ACPS test language is used to compute the test overhead. These overhead measurements can then be used to set variables in the test support software so that test loop overhead (except for operating system statistics) can be accounted for directly in each test execution. Table C-14 shows the command procedure which executes the program AFIRST which is the Ada version of the overhead test. Table C-15 contains output from execution of AFIRST. The output contains 20 executions (ADAFRST) of nonmicrosecond level tests and 10 executions of microsecond level tests (TOVR). The format of the output is described in Appendix E. Based on data from Table C-15, the following initial values would be given to variables in DDEC:ADASY.S.ADA:

- a. For microsecond level tests, the test loop overhead time is contained in variable TOVR. It would be initialized to 213.
- b. For nonmicrosecond level tests:
 1. LOOPN contains the number of test loop iterations. It would be set to 99998.
 2. LOOPTM contains the measured elapsed time for LOOPN iterations through the test loop. It is set to the median of values measured which would be 3.299988.
 3. LOOPZ contains the size in bytes of the test loop compiled code. The preset value for DEC VAX Ada is 0 since the compiler does not support the address attribute for labels.

C.8 DETERMINATION OF TEST REPEATABILITY

As a result of the coding and command procedure conventions adopted in the VAX/VMS ACPS, the elapsed time and CPU time measurements made at run-time can be made accurate to within ± 10 ms. To achieve a test measurement accuracy of 99%, one needs only to ensure that each test (exclusive of test overhead) executes for at least 1 sec.

The special command procedure DDEC:REPEAT, shown in Table C-16, is used in the ACPS to verify measurement accuracy by repeatedly executing the test loop overhead program AFIRST so that 1000 test measurements are made. The run-time test statistic file is then manually inspected to determine the variance in test measurements. If the minimum and maximum measurements differ by more than 20 ms, then to achieve a test measurement accuracy of 99%, the ACPS test durations must be increased accordingly [e.g., for a 40 ms variance, tests must be executed (exclusive of test loop overhead) for at least 3 sec].

Typically, these large test measurement variances are caused by scheduled events within VMS (e.g., VAX cluster and DECNET polling activities or periodic processing by the system process ERRFMT). They can be eliminated by restriction of the VMS software environment. Table C-15 shows an example of a test measurement (3.349976) which was impacted by VMS batch processing logic that flushes batch output buffers at 1 min intervals even if they are empty. ACPS command procedures were subsequently modified to set this interval to one day.

For VAX computers with either large memory configurations (greater than 16 Mbytes) or which have memory modules that have different access times, test measurements cannot be made to be repeatable through varying test durations. Test measurement variances greater than 30% have been obtained in such configurations. These variances are due to the effect of instruction pipelining in which multiple memory accesses may be made simultaneously. The efficiency of memory access for these configurations is dependent upon where programs are located in physical memory. Since the user has no control over how his or her program is loaded into physical memory, any tests run in these configurations are not repeatable and the user must measure the variance of execution time for all tests executed as discussed in Section C.10.

C.9 COMPILATION AND EXECUTION OF ACPS TESTS

Several command procedures are used to compile and execute ACPS tests. As shown in Table C-1, for each common language test (not including type C, E, S, or T tests) functionally equivalent command procedures of similar form are used for each ACPS test language. The batch procedure to compile and execute tests for DEC VAX Ada compiler is shown in Table C-17. It invokes two command procedures: DDEC:ACOMP to compile ACPS tests and DDEC:AEXEC to execute ACPS tests. Compile-time statistics are written to the file DRESULT:ACOMP.DAT and run-time statistics are written to the file DRESULT:AEXEC.DAT.

Table C-18 shows a shortened version of the command procedure DDEC:ACOMP which was used to compile selected ACPS VAX Ada tests for demonstration of ACPS compile-time output which is described in Appendix E. The command procedure is divided into five sections: compilation of tests (lines 44-83), creation of executable modules (lines 86-92), determination of source file size (lines 96-114), determination of object module size (lines 117-134), and determination of executable module size (lines 137-144). The command procedure generates start and end records (e.g., lines 44 and 86) on the output statistic file for use by the ACPS comparison tool to identify where each of these test output sections begins and ends. The symbol < is used to indicate the start of a test record or test section. ACOMP uses the command procedure DDEC:ACOM to compile and generate test output records for single compilation units. As shown in Table C-19, ACOM first uses the symbol PMSG to write a test identification record and then uses the DCL SHOW STATUS command (invoked by symbol SHW defined by DSUP:SYMBOLS) to measure system resource use before and after the compilation. ACOMP uses the command procedure DDEC:ALNK, shown in Table C-20, to form executable modules for single Ada programs and to generate test records showing the resources consumed. ALNK generates test identification and performance statistic records in the same way as ACOM does. ACOMP uses the following command procedures to calculate and display

the size of VAX Ada test files: DDEC:ASRC, shown in Table C-21, for source files; DDEC:AOBJ, shown in Table C-22, for object files; and DDEC:AEEXE, shown in Table C-23, for executable files. These command procedures set global symbols (source_size, object_size, executable_size) that are used by ACOMP for output of the total size of each type of Ada file.

Table C-24 shows a shortened version of the command procedure DDEC:AEEXEC which was used to execute selected ACPS tests compiled by VAX Ada for demonstration of ACPS run-time statistic output which is described in Appendix E. The DCL SHOW STATUS command is used to measure the execution and program loading time of the test loop overhead test (aa00000) and of the entire test suite.

The ACPS test comparison tool assumes that test records within compile-time and run-time test statistic files appear in a specific sorted order which is described in Appendix E Section E.3. All test compilation and execution command procedures provided with the VAX/VMS version of the ACPS comply with this ordering.

C.10 EXECUTION OF ACPS TEST RESULT COMPARISON TOOL

The ACPS test result comparison tool consists of two separate programs: one to compare compile-time test results and one to compare run-time execution results.

The command procedures DTOOL:CCOMP is used to interactively execute the ACPS compile-time test result comparator. Table C-25 contains a listing of DTOOL:CCOMP which describes how to invoke the command procedure, the logical names created, and the filename and filetype conventions used. To compare compile-time test results generated through execution of ACPS command procedures provided for ECSPO JOVIAL and VAX Ada, one would type the following:

```
$ @DTOOL:CCOMP DRESULT:ACOMP DRESULT:JCOMP DRESULT:COMPILE_A_J
```

The comparator tool prompts the user for information to control the comparison. This input is described in Appendix E. At the end of execution the following files would be created:

DRESULT:COMPILE_A_J.U09	-	Output comparing VAX Ada and ECSPO JOVIAL compile-time test statistic files.
DRESULT:COMPILE_A_J.U10	-	Formatted output for VAX Ada compile-time test statistic input file.
DRESULT:COMPILE_A_J.U11	-	Formatted output for ESPO JOVIAL compile-time test statistic input file.

The command procedure DTOOL:CEXEC is used to interactively execute the ACPS run-time test result comparator. The comparator tool prompts the user for information to control processing. This input is described in Appendix E. The listing for DTOOL:CEXEC is contained in Table C-26. It describes how the command procedure is invoked, which logical names it uses, and the filename and filetype conventions employed. To compare run-time test results from execution of ACPS command procedures for ECSP0 JOVIAL and VAX Ada, one would enter the following command:

```
$ @DTOOL:CEXEC DRESULT:AEEXEC DRESULT:JEXEC DRESULT:RUN_A_J
```

At the end of execution, CEXEC would generate the following files:

```
DRESULT:RUN_A_J.U09 - Output comparing VAX Ada and ECSP0 JOVIAL run-time
                      test statistic files.

DRESULT:RUN_A_J.U10 - Formatted output for VAX Ada run-time test statistic
                      input file.

DRESULT:RUN_A_J.U11 - Formatted output for ECSP0 JOVIAL run-time test
                      statistic input file.
```

To measure test repeatability, the execution comparator tool can be used to calculate the maximum and minimum execution statistics for each test. The tool can optionally compare results of successive executions of ACPS test execution command procedures and generate maximum or minimum test statistic files (see lines 42-45 in Table C-26). These test statistic files can be continually updated by comparison with subsequent test executions until the maximum and minimum statistic values for every test do not change. The resultant maximum or minimum test statistic files could then be used for comparison to corresponding maximum or minimum test result files for other test types.

Table C-1. ACPS VAX/VMS Command Procedures

Description	FILE NAMES		
	VAX Ada	ECSPD JOVIAL	VAX FORTRAN
Display VAX/VMS Software Environment	BACPS:VAXVMS	*	*
Define Logical Names and Environment Symbols	BACPS:LOGICALS	*	*
Install Common Language Support Software	DSUP:SUPPORT	*	*
Install ACPS Test Comparison Tools	BTOL:TOOLS	*	*
Setup Language Test Environment	BDEC:SETUP	BJOV:SETUP	BFOR:SETUP
Define Common Language Command Symbols	DSUP:SYMBOLS	*	*
Execute Program and Record Test Results	DSUP:RUN	*	*
Execute Program	DSUP:EXEC	*	*
Execute Show Command and Record Results	DSUP:SHOW	*	*
Execute Show Command	DSUP:SHOW	*	*
Display Ada Compiler Dependent Parameters	BDEC:ADAPARM		
Compile/Execute Test Loop Overhead Test			
- common language tests	BDEC:AFIRST	BJOV:FIRST	BFOR:FFIRST
- Ada type C tests	BDEC:CFIRST		
- Ada type E tests	BDEC:EFIRST		
- Ada type S tests	BDEC:SFIRST		
- Ada type T tests	BDEC:TFIRST		
Determine Run Time Measurement Repeatability	BDEC:REPEAT		
Compile/Execute ACPS tests			
- common language tests	BDEC:ABATCH	BJOV:JBATCH	BFOR:FBATCH
- Ada type C tests	BDEC:CBATCH		
- Ada type E tests	BDEC:EBATCH		
- Ada type S tests	BDEC:SBATCH		
- Ada type T tests	BDEC:TBATCH		
Compile ACPS tests			
- common language tests	BDEC:ACOMP	BJOV:JCOMP	BFOR:FCOMP
- Ada type C tests	BDEC:CCOMP		
- Ada type E tests	BDEC:ECOMP		
- Ada type S tests	BDEC:SCOMP		
- Ada type T tests	BDEC:TCOMP		
Compile Individual ACPS Files	BDEC:ACOM	BJOV:JCOM	BFOR:FCOM
Link ACPS Programs	BDEC:ALNK	BJOV:JLNK	BFOR:FLNK
Determine Size of ACPS Source Files	BDEC:ASRC	BJOV:JSRC	BFOR:FSRC
Determine Size of ACPS Object Files	BDEC:AOBJ	BJOV:JOB	BFOR:FOBJ
Determine Size of ACPS Executable Files	BDEC:AEXE	BJOV:JEXE	BFOR:FEXE
Execute ACPS Tests			
- common language tests	BDEC:AXEC	BJOV:JEXEC	BFOR:FEEXEC
- Ada type C tests	BDEC:CEXEC		
- Ada type E tests	BDEC:EEXEC		
- Ada type S tests	BDEC:SEXEC		
- Ada type T tests	BDEC:TEXEC		
Execute ACPS Compilation Comparator	BTOL:CCOMP	*	*
Execute ACPS Test Execution Comparator	BTOL:CEXEC	*	*
* - common for all languages			

Table C-2. Command Procedure DACPS:LOGICALS.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:
4  $! File: LOGICALS.COM
5  $!
6  $! REK 2/1/88
7  $!
8  $! Description:
9  $!
10 $! This command procedure is used to define logical names for
11 $! ACPS Ada/FORTRAN/JOVIAL test directories and is invoked as follows:
12 $!
13 $! @logicals p1 p2
14 $!
15 $! where:
16 $!
17 $! p1 - is the device containing the ACPS test suite
18 $!      (e.g. sys$disk)
19 $! p2 - is the name of the ACPS test suite top level
20 $!      directory (e.g. user.acps)
21 $!
22 $! At the end of execution the default directory is set to the top
23 $! level directory of the ACPS test suite and the batch output
24 $! rate is set to 1 day.
25 $!
26 $!
27 $! assign logical directory names
28 $!
29 $ assign 'p1':['p2'] dacps
30 $ assign 'p1':['p2'.ada] dada
31 $ assign 'p1':['p2'.fortran] dfor
32 $ assign 'p1':['p2'.jovial] djov
33 $ assign 'p1':['p2'.tools] dtool
34 $ assign 'p1':['p2'.support] dsup
35 $ assign 'p1':['p2'.data] data
36 $ assign 'p1':['p2'.ada.telesoft] dtele
37 $ assign 'p1':['p2'.ada.dec] ddec
38 $ assign 'p1':['p2'.ada.als] dals
39 $ assign 'p1':['p2'.results] dresult
40 $!
41 $! define symbols to set up compiler specific test environments
42 $!
43 $ use_als == @dals:setup.com
44 $ use_fortran == @dfor:setup.com
45 $ use_ddec == @ddec:setup.com
46 $ use_jovial == @djov:setup.com
47 $ use_ttelesoft == @dtele:setup.com
48 $!
49 $! set default directory to top level acps directory and set batch
50 $! output rate to 1 day
51 $!
52 $ set def dacps
53 $ set output_rate=1-0:0:0
54 $!

```

Table C-3 Command Procedure DSUP:SUPPORT.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation
2  $!
3  $! Test:
4  $! File:      SUPPORT.COM
5  $!
6  $!      REK    2/1/88
7  $!
8  $! Description:
9  $!
10 $! **** compile/assemble Ada/FORTRAN/JOVIAL test support software
11 $! **** create object module library file SUPPORT.OLB to contain the
12 $! **** test support software object modules.
13 $!
14 $!      assumed logical names:
15 $!
16 $!      dsup - directory containing test support software
17 $!
18 $ for/check=bounds gettim
19 $ for/check=bounds gstats
20 $ macro geticr
21 $!
22 $ library/object/create support.olb gettim,gstats,geticr
23 $!

```

Table C-4 Command Procedure DT00L:TOOLS.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation
2  $!
3  $! Test:
4  $! File:      TOOLS.COM
5  $!
6  $!          REK   2/1/88
7  $!
8  $! Description:
9  $!
10 $! ***** compile/link Ada/FORTRAN/JOVIAL compilation, execution result
11 $! ***** comparison tools
12 $! ***** The current directory is changed to dtool and left there on exit
13 $!
14 $!          assumed logical names:
15 $!
16 $!          dtool - directory containing comparison tool source,
17 $!                  object, executable files and execution procedures
18 $!
19 $ set def dtool
20 $ for/check=bounds ccomp
21 $ for/check=bounds cexec
22 $ link ccomp
23 $ link cexec
24 $!

```

Table C-5 Command Procedure DACPS:VAXVMS.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test: ..... $!
4  $! File: VAXVMS.COM $!
5  $!
6  $! REK 2/1/88 $!
7  $!
8  $! Description: $!
9  $!
10 $! This command procedure displays VAX/VMS system and process parameters
11 $! which define the software environment under which ACPS tests are
12 $! compiled and executed. $!
13 $!
14 $! The working set parameters are set and displayed separately by each
15 $! ACPS compilation and execution command procedure. $!
16 $!
17 $! display system processes
18 $!
19 $ show system
20 $!
21 $! display process/quotas
22 $!
23 $ show process/quotas
24 $!
25 $! display RMS multi-block and multi-buffer defaults for FORTRAN I/O
26 $!
27 $ show rms_default
28 $!
29 $! display system parameters
30 $!
31 $ run sys$system:sysgen
32 show/all
33 show/special
34 exit

```


Table C-6 Command Procedure DDEC:ADAPARM.COM

```

1  Ada Real Time/Run Time Environment Test      Aerospace Corporation
2  $!
3  $!      Test:
4  $!      File:      ADAPARM.COM
5  $!
6  $!      REK      2/1/88
7  $!
8  $!      Description:
9  $!
10 $!      **** batch procedure to compile/link/execute Ada ACPS test ADAPARM
11 $!      **** output will be written to data set dresult:adaparm.dat
12 $!
13 $!
14 $!      assumptions:
15 $!
16 $!          assumed logical names:
17 $!
18 $!              dacps - top level directory of ACPS test suite
19 $!              dresult - directory that contains ACPS test result output
20 $!              dsup - directory containing common language support
21 $!                      software and command procedures
22 $!
23 $!          assumed symbol values:
24 $!
25 $!              acps - executes command procedure dacps:logicals
26 $!                      to define logical names, global symbols and
27 $!                      set the default directory to dacps:
28 $!
29 $!              aopt - Ada compiler execution options
30 $!
31 $!              asys - list of object modules common to all tests
32 $!
33 $!              our_run - invokes dsup:run.com which executes the
34 $!                      program in the parameter field and appends
35 $!                      the test results to the file with logical
36 $!                      name results.
37 $!
38 $!              use_dec - invokes a command procedure that defines
39 $!                      global symbols used to compile and execute
40 $!                      test programs and sets the current
41 $!                      directory to the one containing
42 $!                      the Ada ACPS test programs
43 $!
44 $!          assumed program library directory: [.lib]
45 $!
46 $!          other assumptions:
47 $!              - support routines have been compiled, assembled
48 $!
49 $!      set up DEC Ada test environment
50 $!
51 $!      acps
52 $!      use_dec
53 $!      show system
54 $!      ----- delete extraneous files -----
55 $!      del [.lib]X.X;X
56 $!      acs create library [.lib]
57 $!      acs set library [.lib]
58 $!      -----
59 $!
60 $!      ----- compile/link test programs -----
61 $!
62 $!      set work /quota=4096/extent=8152/adjust
63 $!      show work
64 $!      @acom adaparm
65 $!      acs link/nomap/command=adaparm.tmp adaparm
66 $!      @adaparm.tmp
67 $!      del adaparm.tmp;X
68 $!
69 $!      ----- run adaparm -----
70 $!
71 $!      set work /noadjust/quota=850/extent=850
72 $!      show work

```

Table C-6 Command Procedure DDEC:ADAPARM.COM (concluded)

```
68      # assign dresult:adaparm.dat results
69      # copy dresult:empty.dat results:
70      # our_run adaparm
71      # show system
```

Table C-7 ADAPARM Output for DEC VAX Ada

```

minimun delay time interval = 1.00000E-02
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
ada clock granularity(seconds)= 0.009949
duration'small= 6.10352E-05
system.tick= 1.00000E-02
system.fine_delta= 9.31323E-10
system.min_int=-2147483648
system.max_int= 2147483647
system.max_digits= 33
system.max_mantissa= 31
standard_input file name=SYS$INPUT.;
standard_output file name=SYS$OUTPUT.;
integer'size= 32
integer'first=-2147483648
integer'last= 2147483647
short_integer'size= 16
short_integer'first= -32768
short_integer'last= 32767
short_short_integer'size= 8
short_short_integer'first= -128
short_short_integer'last= 127
positive'size= 31
positive'first= 1
positive'last= 2147483647
float'size= 32
float'digits= 6
float'emax= 84
long_float'size= 64
long_float'digits= 15
long_float'emax= 204
long_long_float'size= 128
long_long_float'digits= 33
long_long_float'emax= 444

```

Table C-8 Command Procedure DDEC:SETUP.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:
4  $! File:  SETUP.COM
5  $!
6  $!      REK   2/1/88
7  $!
8  $! Description:
9  $!
10 $!
11 $! **** defines DEC Ada compile/link global symbols and sets the default
12 $! **** directory to ddec. It also defines global symbols to invoke command
13 $! **** procedures in directory dsup.
14 $!
15 $!      assumed logical names:
16 $!      ddec      - points to directory containing DEC Ada ACPS tests
17 $!      dsup      - points to directory containing test support
18 $!                  object modules
19 $!      sys$share - directory containing a user written system
20 $!                  service to access microsecond level time
21 $!
22 $!      command procedures used:
23 $!      dsup:symbols - defines global symbols our_run,our_show to invoke
24 $!                  command procedures in directory dsup:
25 $!
26 $! define DCI symbols used in Ada/JOVIAL/FORTRAN command procedures
27 $!
28 $! @dsup:symbols
29 $!
30 $! define compile/link symbols
31 $!
32 $! copt :== aopt :== "/nowarn/nolist/nocheck"
33 $! nopt :== aopt :== "/nowarn/nolist"
34 $! sopt :== aopt :== "/nowarn/nolist/optimize=space"
35 $! topt :== aopt :== "/nowarn/nolist/optimize=time"
36 $! anoopt :== aopt :== "/nowarn/nolist/noopt"
37 $! alist :== aopt :== "/nowarn/list"
38 $! aopt :== /nowarn/nolist
39 $! asys :== dsup:support.olb/library,sys$share:uss.olb/library
40 $!
41 $! set the default directory for DEC Ada
42 $!
43 $! sd ddec
44 $! assign [ul5554.acps.ada.dec.results] dresult

```

Table C-9 Command Procedure DSUP:SYMBOLS.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test: .....
4  $! File:      SYMBOLS.COM
5  $!
6  $!      REK   2/1/88
7  $!
8  $! Description:
9  $!
10 $!
11 $! Defines symbols which invoke language independent command procedures
12 $! and which abbreviate commonly used VAX DCL commands. These symbols
13 $! are used in ACPS compile/execute command procedures.
14 $! This procedure is invoked by each language's setup.com command procedure
15 $! which is used to set up compiler specific compile-time/
16 $! run-time test environments
17 $!
18 $!
19 $! assumptions:
20 $!
21 $!      assumed logical names:
22 $!
23 $!      dsup   - directory for Ada/JOVIAL/FORTRAN common support
24 $!               routines/command procedures
25 $!
26 $  our_r$un    == "dsup:run"
27 $  our_s$how   == "dsup:show"
28 $  sd          == set default
29 $  shw         == show status
30 $  pmsg        == write sys$output

```

Table C-10 Command Procedure DSUP:RUN.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $!   Test:      .....
4  $!   File:      RUN.COM
5  $!
6  $!           REK   2/1/88
7  $!
8  $!   Description:
9  $!
10 $!   **** This command procedure is used to execute Ada/JOVIAL/FORTRAN
11 $!   **** main programs and append the test results to the file with
12 $!   **** logical names results: . It uses the command procedure
13 $!   **** dsup:exec.com to do the actual execution. RUN is invoked as
14 $!   **** follows:
15 $!
16 $!   @run pl
17 $!
18 $!   where 'pl' is the Ada/JOVIAL/FORTRAN program to be executed
19 $!
20 $!
21 $!   assumptions:
22 $!
23 $!       assumed logical names:
24 $!
25 $!           results - file to contain test result output
26 $!           dsup    - directory for Ada/JOVIAL/FORTRAN common support
27 $!                   routines/command procedures
28 $!
29 $!
30 $!   execute the program and direct the results to a temporary file
31 $!
32 $!   @dsup:exec/output=dsup:temp.dat 'pl'
33 $!   append dsup:temp.dat results:
34 $!   del dsup:temp.dat;*

```

Table C-11 Command Procedure DSUP:EXEC.COM

```
1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!  
2  $! $!  
3  $! Test: ..... $!  
4  $! File: EXEC.COM $!  
5  $! $! $!  
6  $! REK 2/1/88 $!  
7  $! $!  
8  $! Description: $!  
9  $! $!  
10 $! ***** This command procedure is common to Ada/FORTRAN/JOVIAL tests  
11 $! ***** and is used to execute main programs. It is invoked as follows:  
12 $! $!  
13 $! @exec 'p1'  
14 $! $!  
15 $! this procedure executes program 'p1'  
16 $! $!  
17 $ run 'p1'
```

Table C-12 Command Procedure DSUP:SHOW.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $!   Test:      .....
4  $!   File:      SHOW.COM
5  $!
6  $!           REK   2/1/88
7  $!
8  $!   Description:
9  $!
10 $! This procedure executes a specified show command and appends its output
11 $! to the file with the logical name results: . It is invoked as follows:
12 $!
13 $! @show 'pl'
14 $!
15 $!     where 'pl' is the show command to be executed
16 $!
17 $! assumptions:
18 $!
19 $!     assumed logical names:
20 $!
21 $!         results - file to contain test result output
22 $!         dsup    - directory for Ada/JOVIAL/FORTRAN common support
23 $!                   routines/command procedures
24 $!
25 $!
26 $ @dsup:show_/output=dsup:temp.dat 'pl'
27 $ append dsup:temp.dat results:
28 $ del dsup:temp.dat;*

```


Table C-13 Command Procedure DSUP:SHOW_.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:      .....COM
4  $! File:      SHOW_.COM
5  $!
6  $!      REK   2/1/88
7  $!
8  $! Description:
9  $!
10 $! This procedure executes a specified show command and is invoked as follows:
11 $!
12 $! @show_ 'p1'
13 $!
14 $!      where 'p1' is the show command to be executed
15 $!
16 $!
17 $ show 'p1'

```

Table C-14 Command Procedure DDEC:AFIRST.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:      .....
4  $! File:      AFIRST.COM
5  $!
6  $!          REK    2/1/88
7  $!
8  $! Description:
9  $!
10 $! **** batch procedure to compile/link/execute Ada ACPS test AFIRST
11 $! **** output of AFIRST will be written to file dresult:afirst.dat
12 $!
13 $!
14 $! assumptions:
15 $!
16 $!     assumed logical names:
17 $!
18 $!         dacps - top level directory of ACPS test suite
19 $!         dresult - directory that contains ACPS test result output
20 $!         dsup   - directory for Ada/JOVIAL/FORTRAN common support
21 $!                  routines/command procedures
22 $!
23 $!     assumed symbol values:
24 $!         acps      - executes command procedure dacps:logicals
25 $!                     to define logical names, global symbols and
26 $!                     set the default directory to dacps:
27 $!         aoapt      - Ada compiler execution options
28 $!         asys       - list of object modules common to all tests
29 $!         our_run    - invokes dsup:run.com which executes the
30 $!                     program in the parameter field and appends
31 $!                     the test results to the file with logical
32 $!                     name results.
33 $!         our_show   - invokes dsup:show.com which executes the
34 $!                     show command in the parameter field and
35 $!                     appends the show command output to the file
36 $!                     with the logical name results.
37 $!         use_dec    - invokes a command procedure that defines
38 $!                     global symbols used to compile and execute
39 $!                     test programs and sets the current
40 $!                     directory to the one containing
41 $!                     the Ada ACPS test programs
42 $!
43 $!     assumed program library directory: [.lib]
44 $!
45 $! other assumptions:
46 $!
47 $!         - dsup support routines have been compiled,assembled
48 $!
49 $! set up DEC Ada test environment
50 $!
51 $! acps
52 $! use_dec
53 $! show system
54 $! ----- delete extraneous files -----
55 $! del [.lib]*.*;
56 $! acs create library [.lib]
57 $! acs set library [.lib]
58 $!
59 $! ----- compile test support routines -----
60 $!
61 $! ada'aoapt oursys
62 $! ada'aoapt ourspc.dec ! for non VAX Ada use ourspc.use
63 $! ada'aoapt ourdmp
64 $! ada'aoapt ourtyp.vax
65 $! ada'aoapt adasys.dec ! for non VAX Ada use adasys.use
66 $! ada'aoapt adaspc
67 $! ada'aoapt adadmp

```

Table C-14 Command Procedure DDEC:AFIRST.COM (concluded)

```

68 $ ada'aopt adatyp.vax
69 $ ada'aopt mathfun.dec !for non VAX Ada - redevelop
70 $!
71 $! ----- compile/link test programs -----
72 $!
73 $ @acom afirst
74 $ @alink afirst
75 $!
76 $! ----- run afirst -----
77 $!
78 $ copy dresult:empty.dat dresult:afirst.dat
79 $ assign dresult:afirst.dat results
80 $ set work /noadjust/quota=850/extent=850
81 $ our_show work
82 $ our_run afirst
83 $ show system

```

Table C-15 AFIRST Output for DEC VAX Ada

<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	²					
<adafirst	1.929993	149999	191	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	³					
<adafirst	1.939941	149999	192	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁴					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁵					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁶					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁷					
<adafirst	1.929993	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁸					
<adafirst	1.939941	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	⁹					
<adafirst	1.939941	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹⁰					
<adafirst	1.939941	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹¹					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹²					
<adafirst	1.939941	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹³					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹⁴					
<adafirst	1.929993	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹⁵					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹⁶					
<adafirst	1.929993	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	⁻¹	¹⁷					
<adafirst	1.939941	149999	194	0	0	0	5

Table C-15 AFIRST Output for DEC VAX Ada (concluded)

<adafirst	0.000000	1	0	0	0	0	5
2	-1	18					
<adafirst	1.939941	149999	193	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	-1	19					
<adafirst	1.939941	149999	194	0	0	0	5
<adafirst	0.000000	1	0	0	0	0	5
2	-1	20					
<adafirst	1.949951	149999	194	0	0	0	5
<tovr	0.000109	0	50	45	0	0	5
2	136	119					
<tovr	0.000107	0	50	44	0	0	5
2	146	115					
<tovr	0.000107	0	50	45	0	0	5
2	135	123					
<tovr	0.000106	0	50	45	0	0	5
2	134	116					
<tovr	0.000107	0	50	44	0	0	5
2	144	122					
<tovr	0.000108	0	50	45	0	0	5
2	145	123					
<tovr	0.000108	0	50	45	0	0	5
2	134	115					
<tovr	0.000109	0	50	44	0	0	5
2	150	119					
<tovr	0.000107	0	50	46	0	0	5
2	138	115					
<tovr	0.000110	0	50	45	0	0	5
2	144	125					

Table C-16 Command Procedure DDEC:REPEAT.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:
4  $! File: REPEAT.COM
5  $!
6  $! REK 2/1/88
7  $!
8  $! Description:
9  $!
10 $! This command procedure is used to execute AFIRST 50 times
11 $! The output will be placed on DRESULT:REPEAT.DAT and should be inspected
12 $! to see how the execution time varies for each AFIRST execution
13 $! in order to determine how long each ACPS test should be executed
14 $!
15 $!
16 $! assumptions:
17 $!
18 $!     assumed logical names:
19 $!
20 $!         dacps - top level directory of ACPS test suite
21 $!         ddec - directory containing DEC Ada ACPS tests
22 $!         dresult - directory that contains ACPS test result output
23 $!         dsup - directory containing common language support
24 $!                 software and command procedures
25 $!
26 $!     assumed symbol values:
27 $!         acps - executes command procedure dacps:logicals
28 $!                 to define logical names, global symbols and
29 $!                 set the default directory to dacps:
30 $!         our_run - invokes dsup:run.com which executes the
31 $!                 program in the parameter field and appends
32 $!                 the test results to the file with logical
33 $!                 name results.
34 $!         our_show - invokes dsup:show.com which executes the
35 $!                 show command in the parameter field and
36 $!                 appends the show command output to the file
37 $!                 with the logical name results.
38 $!         use_dec - invokes a command procedure that defines
39 $!                 global symbols used to compile and execute
40 $!                 test programs and sets the current
41 $!                 directory to the one containing
42 $!                 the Ada ACPS test programs
43 $!
44 $!     other assumptions:
45 $!
46 $!         - AFIRST has been compiled/linked
47 $!
48 $! get to appropriate directory
49 $!
50 $! acps
51 $! use_dec
52 $! show system
53 $!
54 $! ensure that the output is written to REPEAT.DAT
55 $!
56 $! copy dresult:empty.dat dresult:repeat.dat
57 $! assign dresult:repeat.dat results
58 $!
59 $! set the working set size appropriately
60 $!
61 $! set work /noadjust/quota=850/limit=850
62 $! our_show work
63 $!
64 $! execute AFIRST 50 times
65 $!
66 $! count = 0
67 $! loop:

```

Table C-16 Command Procedure DDEC:REPEAT.COM (concluded)

```
68 $      our_run AFIRST
69 $      count = count + 1
70 $      if(count .lt. 50) then goto loop
71 $!
72 $ show system
```

Table C-17 Command Procedure DDEC:ABATCH.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:
4  $! File:      ABATCH.COM
5  $!
6  $!      REK    2/1/88
7  $!
8  $! Description:
9  $!
10 $! **** batch procedure to compile/link/execute Ada ACPS test suite
11 $! **** compilation result output will be on file dresult:acomp.dat
12 $! **** execution    result output will be on file dresult:aexec.dat
13 $!
14 $!
15 $! assumptions:
16 $!
17 $!     assumed logical names:
18 $!
19 $!         dacps - top level directory of ACPS test suite
20 $!         dresult - directory that contains ACPS test result output
21 $!
22 $!     assumed symbol values:
23 $!         acps      - executes command procedure dacps:logicals
24 $!                   to define logical names, global symbols and
25 $!                   set the default directory to dacps:
26 $!         use_dec   - invokes a command procedure that defines
27 $!                   global symbols used to compile and execute
28 $!                   test programs and sets the current
29 $!                   directory to the one containing
30 $!                   the Ada ACPS test programs
31 $!
32 $!     other assumptions:
33 $!
34 $!         - dsup support routines have been compiled,assembled
35 $!
36 $!
37 $! set up DEC Ada test environment
38 $!
39 $! acps
40 $! use_dec
41 $! show system
42 $!
43 $! ----- compile/link test programs -----
44 $!
45 $! @acomp/output=dresult:acomp.dat
46 $!
47 $! ----- execute test programs -----
48 $!
49 $! copy dresult:empty.dat dresult:aexec.dat
50 $! assign dresult:aexec.dat results
51 $! @aexec
52 $! show system

```


Table C-18 Command Procedure DDEC:ACOMP.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test: ..... $!
4  $! File:  ACOMP.COM $!
5  $!
6  $!      REK   2/1/88 $!
7  $!
8  $! Description: $!
9  $!
10 $!
11 $! **** compile/link ADA test programs
12 $!
13 $!     assumed logical names:
14 $!         dsup   - ADA/JOVIAL/ADA common support routines
15 $!     assumed symbol values:
16 $!         aopt   - ADA compiler options
17 $!         asys   - list of object modules common to all tests
18 $!         pmsg   - outputs parameter on systoutput
19 $!         shw    - show status
20 $!
21 $!     command procedures used:
22 $!         acom   - compiles program/package modules and
23 $!                 generates show status records
24 $!         alnk   - links Ada programs and generates show status
25 $!                 records
26 $!         asrc   - calculates/prints/sums source module sizes
27 $!         aobj   - calculates/prints/sums object module sizes
28 $!         aexe   - calculates/prints/sums load module sizes
29 $!
30 $!     assumed program library directory:
31 $!         [.lib]
32 $!
33 $! -----
34 $! set work set limits
35 $ set work /quota=4096/extent=8152/adjust
36 $ show work
37 $! ----- delete extraneous files -----
38 $ del [.lib...].*.x;*.x
39 $ acs create library [.lib]
40 $ acs set library [.lib]
41 $ purge *.x
42 $ del *.obj;*.x;*.exe;*.x
43 $! -----
44 $ pmsg "<start compiles"
45 $! compile ADA test programs and support routines
46 $ pmsg "<system"
47 $ shw
48 $ ada'aopt oursys
49 $ ada'aopt ourspc.dec ! for non VAX Ada use ourspc.use
50 $ ada'aopt ourdmp
51 $ ada'aopt ourtyp.vax
52 $ ada'aopt adasys.dec ! for non VAX Ada use adasys.use
53 $ ada'aopt adaspc
54 $ ada'aopt adadmp
55 $ ada'aopt adatyp.vax
56 $ ada'aopt mathfun.dec !for non VAX Ada - redevelop
57 $ shw
58 $ acs create sublibrary/parent=[.lib] [.lib.sub]
59 $ @acom AA0P000
60 $ @acom AA00000
61 $ @acom AA0P000
62 $ @acom AA00000
63 $ @acom AF0P000
64 $ @acom AF05305
65 $ @acom AF05306
66 $ @acom AF05307
67 $ @acom AF05308

```

Table C-18 Command Procedure DDEC:ACOMP.COM (continued)

```

68  $ @acom AG0P000
69  $ pmsg "<AG00008"
70  $ shw
71  $ ada'aopt AG00003
72  $ ada'aopt AG00004
73  $ ada'aopt AG00005
74  $ ada'aopt AG00006
75  $ ada'aopt AG00007
76  $ ada'aopt AG00008
77  $ shw
78  $ @acom AGM0008
79  $ @acom A00P000
80  $ @acom A00505
81  $ @acom A00506
82  $ @alnk AF05308
83  $ pmsg "<end compiles"
84  $!-----
85  $! link programs
86  $ pmsg "<start links"
87  $ acs set library [.lib]
88  $ @alnk AA00000
89  $ @alnk AG00008
90  $ @alnk AGM0008
91  $ @alnk A00506
92  $ pmsg "<end links"
93  $!-----
94  $! source file storage
95  $ source_size == 0
96  $ pmsg "<start source size"
97  $ @asrc "system" oursys,ourdmp,ourtyp,adasys,adadmp,adatyp,mathfun,ourspc,adasp
98  $ @asrc "" AA0P000
99  $ @asrc "" AA00000
100 $ @asrc "system" oursys,ourdmp,adasys,adadmp
101 $ @asrc "" AA0P000
102 $ @asrc "" AA00000
103 $ @asrc "" AF0P000
104 $ @asrc "" AF05305
105 $ @asrc "" AF05306
106 $ @asrc "" AF05307
107 $ @asrc "" AF05308
108 $ @asrc "" AG0P000
109 $ @asrc AG00008 AG00003,AG00004,AG00005,AG00006,AG00007,AG00008
110 $ @asrc "" AGM0008
111 $ @asrc "" A00P000
112 $ @asrc "" A00505
113 $ @asrc "" A00506
114 $ pmsg "<end source size = 'source_size'"
115 $!-----
116 $! object file storage
117 $ object_size == 0
118 $ pmsg "<start object size"
119 $ assign [.lib] objlib
120 $ @aobj "system" oursys,ourdmp,ourtyp,adasys,adadmp,adatyp,mathfun,ourspc,adasp
121 $ @aobj "" AA0P000
122 $ @aobj "" AA00000
123 $ @aobj "" AF0P000
124 $ @aobj "" AF05305
125 $ @aobj "" AF05306
126 $ @aobj "" AF05307
127 $ @aobj "" AF05308
128 $ @aobj "" AG0P000
129 $ @aobj AG00008 AG00003,AG00004,AG00005,AG00006,AG00007,AG00008
130 $ @aobj "" AGM0008
131 $ @aobj "" A00P000
132 $ @aobj "" A00505
133 $ @aobj "" A00506
134 $ pmsg "<end object size = 'object_size'"

```

Table C-18 Command Procedure DDEC:ACOMP.COM (concluded)

```

135 $!-----
136 $! load module file storage
137 $ executable_size == 0
138 $ pmsg "<start executable size"
139 $ @aexe AA00000
140 $ @aexe AF05308
141 $ @aexe AG00008
142 $ @aexe AGM0008
143 $ @aexe A000506
144 $ pmsg "<end executable size = 'executable_size'"
145 $ deassign objlib
146 $!-----

```

Table C-19. Command Procedure DDEC:ACOM.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $!   Test:      .....
4  $!   File:      ACOM.COM
5  $!
6  $!           REK   2/1/88
7  $!
8  $!   Description:
9  $!
10 $! This command procedure compiles DEC Ada ACPS tests
11 $! It is invoked by the DEC Ada ACPS compile/link command procedure
12 $! with a single parameter that identifies the file to be compiled
13 $! It assumes that the source file is contained in a file with 1 of
14 $! 3 different file types which are searched in the following order:
15 $! .ada, .vax, .dec
16 $! It outputs status records to sys$output that show the cpu/io impact
17 $! of the compilation
18 $!
19 $       if f$search(pl + ".ada") .nes. "" then goto cont_2
20 $       if f$search(pl + ".vax") .nes. "" then goto cont_1
21 $       if f$search(pl + ".dec") .nes. "" then goto cont_0
22 $       goto exit
23 $cont_0: file_name = pl + ".dec"
24 $       goto cont_3
25 $cont_1: file_name = pl + ".vax"
26 $       goto cont_3
27 $cont_2: file_name = pl + ".ada"
28 $cont_3:
29 $ PMSG "<'pl'"
30 $ SHW
31 $ ADA'ADPT 'file_name'
32 $ SHW
33 $ exit:

```

Table C-20 Command Procedure DDEC:ALNK.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!                                              $!
3  $! Test:      .....                          $!
4  $! File:      ALNK.COM                        $!
5  $!                                              $!
6  $!      REK    2/1/88                          $!
7  $!                                              $!
8  $! Description:                                $!
9  $!                                              $!
10 $! This command procedure links DEC Ada ACPS tests
11 $! It is invoked by the DEC Ada ACPS compile/link command procedure
12 $! with a single parameter that identifies the main program name
13 $! It outputs status records to sys$output that show the cpu/io impact
14 $! of the link
15 $!
16 $ PMSG "<'P1'"
17 $ SHW
18 $ ACS LINK/NOMAP/COMMAND='P1'.TMP 'P1' 'ASYS
19 $ @'P1'.TMP
20 $ SHW
21 $ DEL 'P.'.TMP;x

```

Table C-21 Command Procedure DDEC:ASRC.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test: ..... $!
4  $! File:   ASRC.COM $!
5  $!
6  $!      REK   2/1/88 $!
7  $!
8  $! Description: $!
9  $!
10 $!
11 $! This command procedure is used by the DEC Ada ACPS compile/link
12 $! command procedure and is invoked as follows:
13 $!
14 $! @asrc p1 p2
15 $!
16 $! The command procedure determines the size of source files for
17 $! filenames specified in argument p2 and outputs a message to
18 $! sys$output as follows:
19 $!
20 $!      <p1 siz
21 $!
22 $!      where siz is the size of the source files
23 $!      for filenames in parameter p2
24 $!      If p1 is null, then the last filename in p2
25 $!      is used for p1.
26 $! The procedure also adds the size of the source files to the global
27 $! symbol source_size.
28 $!
29 $!      total_size = 0
30 $!
31 $! loop:
32 $!      comma_location = f$locate(", ", p2)
33 $!      file_length = f$length( p2 )
34 $!      if comma_location .eq. file_length then goto last_file
35 $!      file_name = f$extract( 0 , comma_location , p2 )
36 $!      if f$search(file_name + ".ada") .nes. "" then goto loop_2
37 $!      if f$search(file_name + ".vax") .nes. "" then goto loop_1
38 $!      if f$search(file_name + ".dec") .nes. "" then goto loop_0
39 $!      goto loop
40 $! loop_0: file_name = file_name + ".dec"
41 $!      goto loop_3
42 $! loop_1: file_name = file_name + ".vax"
43 $!      goto loop_3
44 $! loop_2: file_name = file_name + ".ada"
45 $!      goto loop_3
46 $! loop_3:
47 $!      total_size = total_size + f$file_attributes( file_name , "eof" )
48 $!      p2 = f$extract( comma_location + 1 , f$length( p2 ) , p2 )
49 $!      goto loop
50 $! last_file:
51 $!      file_name = p2
52 $!      if f$search(file_name + ".ada") .nes. "" then goto last_2
53 $!      if f$search(file_name + ".vax") .nes. "" then goto last_1
54 $!      if f$search(file_name + ".dec") .nes. "" then goto last_0
55 $!      goto exit
56 $! last_0: file_name = file_name + ".dec"
57 $!      goto last_3
58 $! last_1: file_name = file_name + ".vax"
59 $!      goto last_3
60 $! last_2: file_name = file_name + ".ada"
61 $!      goto last_3
62 $! last_3:
63 $!      total_size = total_size + f$file_attributes( file_name , "eof" )
64 $!      if p1 .eqs. "" then p1 = p2
65 $!
66 $! exit:
67 $!      write sys$output "('p1' 'total_size'"
68 $!      source_size == source_size + total_size

```

Table C-22 Command Procedure DDEC:AOBJ.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $!   Test:      .....
4  $!   File:      AOBJ.COM
5  $!
6  $!       REK    2/1/88
7  $!
8  $!   Description:
9  $!
10 $!
11 $! This command procedure is used by the DEC Ada ACPS compile/link
12 $! command procedure and is invoked as follows:
13 $!
14 $! @aoj p1 p2
15 $!
16 $! The command procedure determines the size of object files for
17 $! filenames specified in argument p2 and outputs a message to
18 $! syst$output as follows:
19 $!
20 $!       <p1 siz
21 $!
22 $!           where siz is the size of the object files
23 $!           for filenames in parameter p2.
24 $!           If p1 is null, then the last filename in p2
25 $!           is used for p1.
26 $! The procedure also adds the size of the object files to the global
27 $! symbol object_size.
28 $!
29 $!   assumed logical names:
30 $!       objlib - points to the directory containing the object
31 $!       modules for the input filename
32 $!
33 $!       total_size = 0
34 $!
35 $! loop:
36 $!     comma_location = f$locate(", ", p2)
37 $!     file_length = f$length( p2 )
38 $!     if comma_location .eq. file_length then goto last_file
39 $!     temp_name = f$extract( 0 , comma_location , p2 )
40 $!     under = ""
41 $!     $cont_1: file_name = "objlib:" + temp_name + under + ".obj"
42 $!     if f$search(file_name) .eqs. "" then goto cont_2
43 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
44 $!     file_name = "objlib:" + temp_name + under + ".adc"
45 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
46 $!     file_name = "objlib:" + temp_name + under + ".acu"
47 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
48 $!     $cont_2: if( under .eqs. "_" ) then goto cont_3
49 $!     under = ""
50 $!     goto cont_1
51 $!     $cont_3: p2 = f$extract( comma_location + 1 , f$length( p2 ) , p2 )
52 $!     goto loop
53 $! last_file:
54 $!     under = ""
55 $!     $cont_4: file_name = "objlib:" + p2 + under + ".obj"
56 $!     if f$search(file_name) .eqs. "" then goto cont_5
57 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
58 $!     file_name = "objlib:" + p2 + under + ".acu"
59 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
60 $!     file_name = "objlib:" + p2 + under + ".adc"
61 $!     total_size = total_size + f$file_attributes( file_name , "eof" )
62 $!     $cont_5: if( under .eqs. "_" ) then goto cont_6
63 $!     under = ""
64 $!     goto cont_4
65 $!     $cont_6: if p1 .eqs. "" then p1 = p2
66 $!     write syst$output "<'p1' 'total_size'"
67 $!     object_size == object_size + total_size

```

Table C-23 Command Procedure DDEC:AEXE.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:
4  $! File:      AEXE.COM
5  $!
6  $!      REK    2/1/88
7  $!
8  $! Description:
9  $!
10 $!
11 $! This command procedure is used by the DEC Ada ACPS compile/link
12 $! command procedure and is invoked as follows:
13 $!
14 $! @aexe pl
15 $!
16 $! The command procedure determines the size of the load module file
17 $! for the file specified in argument pl and outputs a message to
18 $! sys$output as follows:
19 $!      <pl siz
20 $!           where siz is the size of the load files
21 $!           for the filename in parameter pl
22 $! The procedure also adds the size of the load files to the global
23 $! symbol executable_size.
24 $!
25 $      file_name = "'pl'" + ".exe"
26 $      total_size == f$file_attributes( file_name , "eof" )
27 $      write sys$output "<'pl' 'total_size'"
28 $      executable_size == executable_size + total_size

```


Table C-24 Command Procedure DDEC:AEEXEC.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation $!
2  $!
3  $! Test:      .....COM
4  $! File:      AEEXEC.COM
5  $!
6  $!          REK   2/1/88
7  $!
8  $! Description:
9  $!
10 $!
11 $! this command procedure executes DEC Ada versions of ACPS test programs
12 $!
13 $!     assumed logical names:
14 $!         data   - points to directory containing test input data
15 $!                   files
16 $!         dsup   - directory containing common language support
17 $!                   software and command procedures
18 $!
19 $!     assumed symbol values:
20 $!         our_run - invokes dsup:run.com which executes the
21 $!                   program in the parameter field and appends
22 $!                   the test results to the file with logical
23 $!                   name results.
24 $!         our_show - invokes dsup:show.com which executes the
25 $!                   show command in the parameter field and
26 $!                   appends the show command output to the file
27 $!                   with the logical name results.
28 $!
29 $ set work/limit=850/quota=850/noadjust
30 $ our_show work
31 $!
32 $! use show status to time aa00000 and all test program loading time
33 $!
34 $ our_show status
35 $ our_run AA00000
36 $ our_show status
37 $ our_show status
38 $ our_run af05308
39 $ our_run ag00008
40 $ our_run agm0008
41 $ our_run ao00506
42 $ our_show status
43 $ set work/adjust
44 $ our_show status
45 $ set work/adjust

```

Table C-25. Command Procedure DTOOL:CCOMP.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation c
2  $!
3  $! Test:
4  $! File:      CCOMP.COM                          c
5  $!
6  $!      REK   2/1/88                             c
7  $!
8  $! Description:                                  c
9  $!
10 $! This file contains the command procedure to execute the ACPS
11 $! compilation comparison program CCOMP and is invoked as follows:
12 $!
13 $! @ccomp p1 p2 p3 p4
14 $!
15 $!      where
16 $!
17 $!      p1   - gives the filename of the compilation result file
18 $!             assigned to logical unit 7
19 $!      p2   - gives the filename of the compilation result file
20 $!             assigned to logical unit 8
21 $!      p3   - gives the filename of CCOMP outputs on logical units
22 $!             9,10,11
23 $!      p4   - gives the filename of command input. If absent, then
24 $!             command input is read from the terminal
25 $!
26 $! The following shows logical name assignments and complete file
27 $! names for all input/output files to CCOMP:
28 $!
29 $!      name  type  filename      description
30 $!      ----  ---  -
31 $!
32 $!      for005 input  'p4'.dat      User command input
33 $!      for006 output sys$output   Terminal output for prompts and error
34 $!                                     messages
35 $!      for007 input  'p1'.dat      Unformatted ACPS compilation result file
36 $!      for008 input  'p2'.dat      Unformatted ACPS compilation result file
37 $!      for009 output 'p3'.u09      ACPS compilation comparison file
38 $!      for010 output 'p3'.u10      Unit 7 formatted with headings and with
39 $!                                     VMS status records processed
40 $!      for011 output 'p3'.u11      Unit 8 formatted with headings and with
41 $!                                     VMS status records processed
42 $!
43 $! assumptions:
44 $!
45 $!      assumed logical names:
46 $!
47 $!      dtool - directory containing ACPS compilation/test result
48 $!               comparison tools
49 $!
50 $! if p4 .nes. "" then assign 'p4'.dat for005
51 $! if p4 .eqs. "" then assign sys$command for005
52 $! assign 'p1'.dat for007
53 $! assign 'p2'.dat for008
54 $! assign 'p3'.u09 for009
55 $! assign 'p3'.u10 for010
56 $! assign 'p3'.u11 for011
57 $! assign sys$output for006
58 $! run dtool:ccomp
59 $! deassign for005
60 $! deassign for006
61 $! deassign for007
62 $! deassign for008
63 $! deassign for009
64 $! deassign for010
65 $! deassign for011

```

Table C-26. Command Procedure DTOOL:CEXEC.COM

```

1  $! Ada Real Time/Run Time Environment Test      Aerospace Corporation c
2  $!
3  $! Test:
4  $! File:      CEXEC.COM                          c
5  $!
6  $!      REK    2/1/88                            c
7  $!
8  $! Description:                                  c
9  $!
10 $! This file contains the command procedure to execute the ACPS
11 $! test result comparison program CEXEC and is invoked as follows:
12 $!
13 $! @ccomp p1 p2 p3 p4
14 $!
15 $!      where
16 $!
17 $!      p1    - gives the filename of the test result file to be
18 $!               assigned to logical unit 7
19 $!      p2    - gives the filename of the test result file to be
20 $!               assigned to logical unit 8
21 $!      p3    - gives the filename of CEXEC outputs on logical units
22 $!               9,10,11,12
23 $!      p4    - gives the filename of command input. If absent, then
24 $!               command input is read from the terminal
25 $!
26 $! The following shows logical name assignments and complete file
27 $! names for all input/output files to CEXEC:
28 $!
29 $!      name      type      filename      description
30 $!      ----      -
31 $!
32 $!      for005    input    'p4'.dat      User command input
33 $!      for006    output    systoutput    Terminal output for prompts and error
34 $!                                     messages
35 $!      for007    input    'p1'.dat      Unformatted ACPS test result file
36 $!      for008    input    'p2'.dat      Unformatted ACPS test result file
37 $!      for009    output    'p3'.u09     ACPS test result comparison file
38 $!      for010    output    'p3'.u10     Unit 7 formatted with headings and with
39 $!                                     test overhead values subtracted
40 $!      for011    output    'p3'.u11     Unit 8 formatted with headings and with
41 $!                                     test overhead values subtracted
42 $!      for012    output    'p3'.u12     Optionally generated file that contains
43 $!                                     the minimum or maximum run statistics
44 $!                                     for corresponding test results output on
45 $!                                     units for010,for011
46 $!
47 $!      assumptions:
48 $!
49 $!      assumed logical names:
50 $!
51 $!      dtool - directory containing ACPS compilation/test result
52 $!               comparison tools
53 $!
54 $! if p4 .nes. "" then assign 'p4'.dat for005
55 $! if p4 .eqs. "" then assign systcommand for005
56 $! assign 'p1'.dat for007
57 $! assign 'p2'.dat for008
58 $! assign 'p3'.u09 for009
59 $! assign 'p3'.u10 for010
60 $! assign 'p3'.u11 for011
61 $! assign 'p3'.u12 for012
62 $! assign systoutput for006
63 $! run dtool:cexec
64 $! deassign for005
65 $! deassign for006
66 $! deassign for007
67 $! deassign for008
68 $! deassign for009
69 $! deassign for010
70 $! deassign for011
71 $! deassign for012

```

APPENDIX D

Ada TEST INTERFACE PACKAGE SPECIFICATIONS

This appendix lists the Ada package specifications for the common language test support software. It also lists the Ada specifications for non-Ada procedures that access target operating system-dependent performance statistics. All compiler-dependent statements (i.e., those suffixed with the string `--*`) are specific to the DEC VAX Ada compiler and may have to be changed for other Ada compilers.

Tables D-1 and D-2 list the specifications for the package OURSYS which defines test support procedures, data types, and global variables for use by all Ada tests. Table D-3 lists the specification for the package OURSPC which defines numerous composite data types and global variables that are used by many ACPS tests. Test descriptions in Appendix F refer to these types by name using capital letters (e.g., RECORDS and RECORD_POINTER). Table D-4 lists the specification for the package OURTYP which defines compiler- and machine-dependent numeric data types and global variables declared with these types. Test descriptions in Appendix F refer to these types by name if the test uses any of the types or global variables declared in OURTYP. The package specification for test support software procedures that dump global variables contained in packages OURSYS and OURSPC is shown in Table D-5. The package MATHFUN, shown in Table D-6, is used by tests to access compiler-dependent mathematics function libraries. Rename statements are used to standardize names of mathematics routines to those used in the Whetstone test.

The package body of OURSYS implements the test support procedures defined in the package specification shown in Table D-1. To provide the ability to read time to the microsecond level and to access additional performance statistics, procedures within the body of OURSYS reference operating system-dependent procedures to perform these tasks. The Ada specification for these procedures is shown in Table D-2 which lists the declarative part of the package body for OURSYS. As shown in the interface pragma statements contained in lines 281 to 285 of Table D-2, these procedures are: GETTIM, GSTATS, INITIM, ISTATS, and SECS. For the VAX/VMS version of ACPS, these procedures are provided in the directory [acps.support] as FORTRAN and MACRO subroutines. As can be seen in Appendix A, these routines are not supplied on the ACPS delivery tape in ANSI format. New versions of these routines must be implemented for each new target operating system.

The following provides further clarifications on the function of these external procedures:

- a. Microsecond level time is assumed to be a floating point number expressed in units of seconds to microsecond level accuracy.
- b. The value returned by the external function SECS is assumed to be in floating point format in units of seconds.

- c. The operating system statistic array set by procedure GSTATS is assumed to contain OURSYS.NCPUST (i.e., 9) elements. The first OURSYS.NTOSUB (i.e., 4) elements are assumed to increase and the difference of the values from the start to the end of a test will be output on the test result file. For the remaining elements in the array, the value at the end of test execution is output to the test result file.

Table D-1 OURSYS Package Specification

```

1  -- Ada Real time/Run time Environment Test      Aerospace Corporation --
2  --
3  -- Test:      .....
4  -- File:      OURSYS
5  --
6  --          REK,MJM,KMB    2/1/88
7  --
8  -- Description:
9  --
10 -- **** This package defines the language independent --
11 -- **** test statistic interface routines used by all ACPS tests. --
12 -- **** OURSYS defines various numeric datatypes(e.g our_integer) to --
13 -- **** ensure that for each test the size of datatypes manipulated --
14 -- **** is compiler independent for a single machine (e.g. VAX). Global--
15 -- **** variables of various datatypes are defined for use by each test--
16 -- **** program.
17 -- ****
18 -- **** Note: In the Ada version of OURSYS, statements followed by --X --
19 -- **** are implementation dependent and may need to be changed --
20 -- **** for each compiler and for each machine.
21 -- **** Changes may also be required to the corresponding --
22 -- **** statements in the JOVIAL and FORTRAN versions of OURSYS. --
23 -- ****
24
25 with system; use system;
26 with calendar; use calendar;
27 with text_io; use text_io;
28 with unchecked_conversion;
29 package oursys is
30
31
32 -- ****
33 -- **** data types (machine word length integer, float) used in tests --
34 -- ****
35
36
37
38 subtype our_string is string; --X
39 type our_fixed      is delta 0.125 range 0.0 .. 1024.0;
40
41 subtype our_integer is integer      ;--X
42 subtype our_positive is positive    ;--X
43 subtype our_natural is natural      ;--X
44 subtype our_float   is float        ;--X
45 subtype one_to_3    is integer range 1..3;
46
47 -- file management definitions --
48 subtype file_name   is our_string(1..11) ;--X
49 subtype file_form   is our_string(1..80) ;--X
50 subtype fin_fo      is our_string(1..8)  ;--X
51 subtype test_id     is our_string(1..7)  ;--X
52 subtype test_ext    is our_string(1..3)  ;--X
53
54
55 -- ****
56 -- **** unchecked conversions
57 -- ****
58
59
60 function address_to_integer is new
61     unchecked_conversion(address,our_integer); --X
62
63
64 -- ****
65 -- **** generic I/O instantiations, one per base type --
66 -- ****
67

```

Table D-1 OURSYS Package Specification (continued)

```

68 package ourfixed_io is new fixed_io (our_fixed);
69 package ourfloat_io is new float_io (our_float);
70 package ourinteger_io is new integer_io (our_integer);
71
72 use ourfixed_io;
73 use ourfloat_io;
74 use ourinteger_io;
75
76
77 -- ***** --
78 -- global test timing interface procedures --
79 -- ***** --
80
81 -- finame (testid, fname) - constructs a --
82 -- implementation dependent --
83 -- file name --
84
85 -- fiorm (finfo, fform) - constructs a --
86 -- implementation dependent --
87 -- file form --
88
89
90 -- mstart - start a microsecond level test iteration --
91
92 -- mstop - stop a microsecond level test iteration --
93 -- and record the elapsed time --
94
95 -- nooptm - used to prevent optimization of test code --
96 -- within test loops. Also causes output of test --
97 -- statistics after first test iteration --
98
99 -- tcomp - cause comparison of the next test execution --
100 -- to the saved test execution --
101
102 -- tfail (errcod) - outputs a test failure error message with --
103 -- error code errcod --
104
105 -- tinit - initialize test environment --
106 -- call istats to initialize non-ada routines --
107 -- determine overhead of normal test loop --
108 -- determine overhead of microsecond level test --
109 -- loop --
110
111 -- tprint - print test results stored by tstop and tstopm --
112
113 -- tsave - save test statistics for later comparison --
114
115 -- tstart (td) - start a test using ada time (if possible)
116 -- td - test identification (7 characters)
117
118 -- tstopm - retrieve & store test statistics at end of --
119 -- microsecond tests. --
120
121 -- tstop - retrieve & store test statistics at end of --
122 -- a test --
123
124 -- tlvalu (val) - store val into print buffers
125
126 -- t2valu (tv1, tv2) - stores test values tv1, tv2 into test print --
127 -- buffers for output with test results --
128
129 procedure finame (testid : in test_id;
130 fname : out our_string
131 );
132
133 procedure fiorm (finfo : in finfo;
134 fform : out our_string

```

Table D-1 OURSYS Package Specification (continued)

```

135         );
136
137     procedure mstart;
138
139     procedure mstop;
140
141     procedure nooptm;
142
143     procedure tcomp (testn : test_id);
144
145     procedure tsave;
146
147     procedure tfail (errcod : in our_integer);
148
149     procedure tinit;
150
151     procedure tprint;
152
153     procedure tstart (td : in test_id);
154
155     procedure tstopm;
156
157     procedure tstop;
158
159     procedure tlvalu (val : in our_integer);
160
161     procedure t2valu (vall, val2 : in address);
162
163     -- ***** --
164     -- test iteration parameters--
165     -- ***** --
166
167     -- maxitr - maximum # of loop iterations in tests --
168     -- micitr - maximum # of iterations in microsecond level tests --
169
170     maxitr : constant our_integer := 150000; --X
171     micitr : constant our_integer := 50;
172
173     -- ***** --
174     -- test global variables--
175     -- ***** --
176
177
178     -- all global variables are initialized by tinit --
179     -- global variables are reset by test start procedures --
180     -- (tstart, mstart) and are conditionally set/used by nooptm --
181     -- to prevent removal of test code from the test loops. --
182
183     -- debug - used to control whether print is generated from routines --
184     -- in package ourdmp that are called by test programs to dump --
185     -- global variables for debugging purposes. --
186
187     type debug_type is (OFF,ON);
188     debug : debug_type := OFF;
189
190     -- delay_test - used to indicate in microsecond level tests whether the
191     -- Ada delay statement is being tested
192
193     delay_test : boolean := false;
194
195     -- max_memory - this is the maximum memory unit size used in the
196     -- memory management tests. The largest amount of memory
197     -- created dynamically will be an array of 1 .. max_memory.
198
199     max_memory : constant our_integer := 2**16; --X
200
201

```


Table D-1 OURSYS Package Specification (continued)

```

202  -- minimum_delay_interval - this is the minimum delay time supplied in
203  --                          an Ada delay statement to force invocation
204  --                          of a lower priority task. It is determined
205  --                          automatically by the program ADAPARM.
206
207      minimum_delay_interval : constant duration := 0.01; --X
208
209  -- our_delay - the delay interval used in microsecond level tasking tests
210  --              (e.g. time to wake up after a delay, time to activate
211  --              a lower priority task after a delay). It should
212  --              be greater than the minimum delay interval supported
213  --              by an Ada run time environment and >= duration'small,
214  --              system.tick.
215
216      our_delay : constant duration := 10 * minimum_delay_interval; --X
217
218  -- liters - used by all test loops as the test iteration count--
219  --          initialized by tinit, tstart --
220  --          incremented by nooptm, mstart --
221
222      liters : our_integer := 0;
223
224
225
226  -- set_tmp - used to determine whether mstart and --
227  --          tstart are to reset the tmp* global variables. set_tmp is set by --
228  --          test programs before calling tstart to indicate that initial --
229  --          values are being given to the tmp* variables. tstop and tstopm --
230  --          always reset set_tmp --
231
232      set_tmp : boolean := false;
233
234  -- integer variables ( all variables = 3 except tmpis0 which is 0) --
235  --          set/used --
236
237      tmpis0 : our_integer;
238      tmpis1 : our_integer;
239      tmpis2 : our_integer;
240      tmpis3 : our_integer;
241      tmpis4 : our_integer;
242      tmpis5 : our_integer;
243      tmpis6 : our_integer;
244      tmpis7 : our_integer;
245      tmpis8 : our_integer;
246      tmpis9 : our_integer;
247      tmpisa : our_integer;
248      tmpisb : our_integer;
249      tmpisc : our_integer;
250      tmpisd : our_integer;
251      tmpise : our_integer;
252      tempi : our_integer;
253
254      type max_mem is array(our_integer range <>) of our_integer;
255      type array_pointer is access max_mem;
256      type integer_vector is array (our_integer range <>) of our_integer;
257      type integer_matrix is array (our_integer range <>,
258                                   our_integer range <>) of our_integer;
259      type integer_3d      is array (our_integer range <>,
260                                   our_integer range <>,
261                                   our_integer range <>) of our_integer;
262      subtype integer_array_50 is integer_vector (1..50) ;
263      subtype integer_array_3_3 is integer_matrix (1..3, 1..3) ;
264      subtype integer_array_3_3_3 is integer_3d (1..3, 1..3, 1..3) ;
265      subtype integer_array_10 is integer_vector (1..10) ;
266      tmpia1 : integer_array_50;
267      tmpia2 : integer_array_3_3;
268      tmpia3 : integer_array_3_3_3;

```

Table D-1 OURSYS Package Specification (continued)

```

269      tmpia4,tmpia5,tmpia6 : integer_array_10;
270      mem : array_pointer;
271
272  -- arrays not defined in type statements --
273
274      tmpau1 : array ( our_integer range 1..50) of our_integer;
275      tmpau2 : array ( our_integer range 1..3, our_integer range 1..3)
276                of our_integer;
277      tmpau3 : array ( our_integer range 1..3, our_integer range 1..3,
278                our_integer range 1..3) of our_integer;
279
280  --
281                used only --
282
283      tmpiu1 : our_integer;
284      tmpiu2 : our_integer;
285      tmpiu3 : our_integer;
286      tmpiu4 : our_integer;
287      tmpiu5 : our_integer;
288
289  -- integer variables range 1..3 --
290  --      set/used --
291      tmpi1 : one_to_3;
292      tmpi2 : one_to_3;
293      tmpi3 : one_to_3;
294      tmpi4 : one_to_3;
295      tmpi5 : one_to_3;
296      tmpi6 : one_to_3;
297      tmpi7 : one_to_3;
298      tmpi8 : one_to_3;
299      tmpi9 : one_to_3;
300      tmpia : one_to_3;
301
302  -- positive variables --
303  --      set/used --
304      tmpn1 : our_positive;
305      tmpn2 : our_positive;
306      tmpn3 : our_positive;
307      tmpn4 : our_positive;
308      tmpn5 : our_positive;
309      tmpn6 : our_positive;
310      tmpn7 : our_positive;
311      tmpn8 : our_positive;
312      tmpn9 : our_positive;
313      tmpna : our_positive;
314
315  -- boolean variables --
316  --      set/used --
317      tmpbs1 : boolean;
318      tmpbs2 : boolean;
319      tmpbs3 : boolean;
320      tmpbs4 : boolean;
321      tmpbs5 : boolean;
322      tmpbs6 : boolean;
323      tmpbs7 : boolean;
324      tmpbs8 : boolean;
325      tmpbs9 : boolean;
326      tmpbsa : boolean;
327      tmpbsb : boolean;
328      tmpbsc : boolean;
329      tmpbsd : boolean;
330      tmpbse : boolean;
331      type boolean_vector is array (our_integer range <>) of boolean;
332      type boolean_matrix is array (our_integer range <>,
333                our_integer range <>) of boolean;
334      type boolean_3d      is array (our_integer range <>,
335                our_integer range <>,
336                our_integer range <>) of boolean;

```

Table D-1 OURSYS Package Specification (concluded)

```

336      subtype boolean_array_10 is boolean_vector(1..10) ;
337      subtype boolean_array_3_3 is boolean_matrix(1..3, 1..3) ;
338      subtype boolean_array_3_3_3 is boolean_3d(1..3, 1..3, 1..3) ;
339      tmpba1,tmpba2,tmpba3 : boolean_array_10;
340      tmpba4 : boolean_array_3_3;
341      tmpba5 : boolean_array_3_3_3;
342
343
344      -- fixed point variables --
345      tmpfs0 : our_fixed;
346      tmpfs1 : our_fixed;
347      tmpfs2 : our_fixed;
348      tmpfs3 : our_fixed;
349      tmpfs4 : our_fixed;
350      tmpfs5 : our_fixed;
351      tmpfs6 : our_fixed;
352      tmpfs7 : our_fixed;
353      tmpfs8 : our_fixed;
354      tmpfs9 : our_fixed;
355      tmpfsa : our_fixed;
356
357
358      -- float variables --
359      --      set/used --
360      tmprs0 : our_float;
361      tmprs1 : our_float;
362      tmprs2 : our_float;
363      tmprs3 : our_float;
364      tmprs4 : our_float;
365      tmprs5 : our_float;
366      tmprs6 : our_float;
367      tmprs7 : our_float;
368      tmprs8 : our_float;
369      tmprs9 : our_float;
370      tmprsa : our_float;
371      tmprsb : our_float;
372      tmprsc : our_float;
373      tmprsd : our_float;
374      tmprse : our_float;
375      tempr : our_float;
376      type float_vector is array (our_integer range <>) of our_float;
377      type float_matrix is array (our_integer range <>,
378                                our_integer range <>) of our_float;
379      type float_3d      is array (our_integer range <>,
380                                our_integer range <>,
381                                our_integer range <>) of our_float;
382      subtype float_array_3_3 is float_matrix (1..3, 1..3) ;
383      subtype float_array_3_3_3 is float_3d (1..3, 1..3, 1..3) ;
384      subtype float_array_10 is float_vector (1..10) ;
385      tmpra1,tmpra2,tmpra3 : float_array_10;
386      tmpra4 : float_array_3_3;
387      tmpra5 : float_array_3_3_3;
388
389      tempf : our_fixed;
390
391      -- string variables --
392      --      set/used --
393      tmpcs1 : our_string(1..10);
394      tmpcs2 : our_string(1..20);
395      tmpcs3 : our_string(1..30);
396      tmpcs4 : our_string(1..10);
397      tmpcs5 : our_string(1..10);
398
399      end oursys;

```

Table D-2. OURSYS Package Body Declarations

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test: .....
4  -- File:  ADASYS
5  --
6  --      REK,MJM,KMB    2/1/88
7  --
8  -- Description:
9  --
10 -- **** This file contains the package body for OURSYS. It implements --
11 -- **** test statistic interface routines used by all ACPs tests. It --
12 -- **** references machine dependent routines ( see pragma interface --
13 -- **** statements within the package body) to access machine dependent--
14 -- **** performance statistics and read time to the microsecond level. --
15 -- ****
16 -- **** Note: In the Ada version of OURSYS, statements followed by --X --
17 -- **** are implementation dependent and may need to be changed --
18 -- **** for each compiler and for each machine.
19 -- **** Changes may also be required to the corresponding
20 -- **** statements in the JOVIAL and FORTRAN versions of OURSYS. --
21 -- ****
22
23 with ourdmp; use ourdmp;
24 with ourtyp; use ourtyp;
25 with ourspc; use ourspc;
26
27 package body oursys is
28 --
29 -- ****
30 -- data types used in the package body of oursys --
31 -- ****
32 --
33     subtype our_duration is our_float      ;--X
34     subtype our_micro    is our_float      ;--X
35     subtype our_stat      is integer        ;--X
36     subtype our_time is time; --X
37
38 -- ****
39 -- machine dependent test parameters
40 -- ****
41
42 -- ntosub - # of statistics returned by gstats that always increase --
43 -- and for which the difference (end-start) will be output --
44 -- ncpust - # of machine statistics set by gstats --
45 -- mintim - minimum test period in seconds --
46 -- tclock - type of clock (Ada, operating system) to be read in --
47 -- non-microsecond level tests. --
48 -- If the operating system clock is used, then the subtype --
49 -- our_time should be set appropriately (for VAX/VMS, --
50 -- use our_float) and the function wclock modified if necessary --
51
52     ncpust : constant our_integer := 9; --X
53     type mach_stats is array (our_integer range 1..ncpust) of our_stat; --X
54
55     ntosub : constant our_integer := 4; --X
56     mintim : constant our_duration := 1.0; --X
57
58     type clock_type is (Ada, OS);
59     tclock : constant clock_type := Ada; --X
60
61
62 -- ****
63 -- local test timing interface variables and procedures --
64 -- ****
65
66
67 -- cstore - stores comparison of previous/saved test results --
68 -- for optimization tests in which testcp(3,3) = C --
69 -- The comparison is done by division --

```

Table D-2. OURSYS Package Body Declarations (Continued)

```

70
71 -- sstore      - stores comparison of previous/saved test results --
72 --              for tests in which testcp(3:3) = S. --
73 --              The comparison is done by subtraction --
74
75 -- stotal      - stores test results in totX arrays --
76
77 -- tmpref      - called by nooptm to reference global variables --
78
79 -- tmpset      - called by test start procedures to set all --
80 --              set/used global variables made visible by oursys --
81
82 -- tstore      - store test results. calls gstats to get machine --
83 --              dependent statistics and stotal to store results --
84
85 -- gettim (tm)  - non-Ada procedure that stores microsecond level --
86 --              time into tm --
87
88 -- gstats (stary,retcod) - non-Ada procedure which acquires non-Ada--
89 --              run statistics and stores them into array stary --
90 --              retcod is a status return code. The value 1 --
91 --              signifies success --
92
93 -- initim      - non-Ada procedure that stores an initial time for --
94 --              microsecond level tests so that test durations --
95 --              can be computed to microsecond level accuracy --
96
97 -- istats      - non-Ada procedure to initialize non-Ada run --
98 --              statistics --
99
100 -- secs        - non-Ada function that returns time in seconds --
101 --              it is used if the value OS is assigned to tclock --
102
103 -- wclock      - returns time (Ada, OS) based on value of tclock --
104
105 procedure cstore;
106
107 procedure sstore;
108
109 procedure stotal;
110
111 procedure tmpref;
112
113 procedure tmpset;
114
115 procedure tstore;
116
117 procedure gettim (tm : out our_micro);
118
119 procedure gstats (stary : out mach_stats; retcod : out our_integer);
120
121 procedure initim;
122
123 procedure istats;
124
125 function secs return our_time;
126
127 function wclock return our_time;
128
129
130 --          individual test statistic variables  --
131
132 -- testc - flag if set signals that the next test is to be compared --
133 --         with a saved test --
134
135 -- testcp - name of test that compares results of two previous tests --
136
137 -- testnm - name of test --
138
139 -- initok - set if time related routines have been executed once --

```

Table D-2. OURSYS Package Body Declarations (Continued)

```

140
141 -- niters - counter for number of loop iterations executed in a test --
142
143 -- niter1 - contains niters for the previous test --
144
145 -- niter2 - contains niters for the saved test --
146
147 -- once - indicates whether test code has been executed at least once --
148
149 -- ntstat - # of possible entries in tstatv --
150
151 -- tstat1 - index to last entry stored into tstatv --
152
153 -- tstatv - print buffer for test values inserted by t2valu&tlvalu --
154
155 -- tstat1 - contains tstat1 for the previous test --
156
157 -- tstat2 - contains tstat1 for the saved test --
158
159 -- tstatv - contains tstatv values for the previous test --
160
161 -- tstatb - contains tstatv values for the saved test --
162
163 -- strtim - time at start of test --
164
165 -- endtim - time at end of test --
166
167 -- strmic - microsecond level time at start of test --
168
169 -- endmic - microsecond level time at end of test --
170
171 -- elaptm - elapsed time of test in seconds --
172
173 -- elapt1 - elapsed time for the previous test --
174
175 -- elapt2 - elapsed time for the saved test --
176
177 -- micbuf - buffer to hold microsecond elapsed times for a test --
178
179 -- micind - index to last entry stored into micbuf --
180
181 -- loopn - # of iterations in loop timed by loopn --
182
183 -- loopsz - size of test loop overhead code(bytes) --
184
185 -- looptm - elapsed time overhead of test loop for loopn iterations --
186
187 -- tovr - elapsed time overhead in calling gettim for microsecond --
188 -- level tests --
189
190 -- ststat - machine dependent statistics at start of test --
191
192 -- estat - machine dependent statistics at end of test --
193
194 -- estat1 - machine test statistics of the previous test --
195
196 -- estat2 - machine test statistics of the saved test --
197
198
199 initok : boolean := false;
200 once : boolean := false;
201 testc : boolean;
202
203 testcp : test_id;
204 testnm : test_id;
205
206 micind : our_integer := 0;
207 niters : our_integer := 0;
208 niter1 : our_integer := 0;
209 niter2 : our_integer := 0;

```

Table D-2. OURSYS Package Body Declarations (Continued)

```

210  tstati : our_integer := 0;
211  tstat1 : our_integer;
212  tstat2 : our_integer;
213
214  ntstat : constant our_integer := 5;
215  tstatv : array (our_integer range 1..ntstat) of our_integer;
216  tstat : array (our_integer range 1..ntstat) of our_integer;
217  tstatb : array (our_integer range 1..ntstat) of our_integer;
218
219  strtim : our_time;
220  endtim : our_time;
221
222  strmic : our_micro;
223  endmic : our_micro;
224
225  elaptm : our_duration;
226  elapt1 : our_duration;
227  elapt2 : our_duration;
228
229  loopn : our_integer := 0;      -- preset determined by AFIRST --
230  loopnz : our_integer := 0;    -- preset determined by AFIRST --
231  loopm : our_duration := 1.665000; -- preset determined by AFIRST --
232  tovr : our_duration := 0.000000; -- preset determined by AFIRST --
233
234  type tvector is array (our_integer range 1..micitr) of our_duration;
235  micbuf : tvector;
236
237  sstat : mach_stats;
238  estat : mach_stats;
239  estat1 : mach_stats;
240  estat2 : mach_stats;
241
242  -- accumulation test statistic variables --
243
244  -- ntot   - # of test results saved before printing --
245  --        - used as a dimension limit in totX arrays --
246
247  -- totind - index of last entry in totX arrays --
248
249  -- totelp - contains elaptm for all tests run --
250
251  -- totitr - contains niters for all tests run --
252
253  -- totmac - contains estat for all tests run --
254
255  -- totnm  - contains testnm for all tests run --
256
257  -- totrun - contains tstatv for all tests run --
258
259  -- totsti - contains tstati for all tests run --
260
261  ntot : constant our_integer := 75;
262
263  totind : our_integer := 0;
264
265  totelp : array (our_integer range 1..ntot) of our_duration;
266  totitr : array (our_integer range 1..ntot) of our_integer;
267  totmac : array (our_integer range 1..ncpust, our_integer range 1..ntot)
268  of our_stat;
269  totnm : array (our_integer range 1..ntot) of test_id;
270  totrun : array (our_integer range 1..ntstat, our_integer range 1..ntot)
271  of our_integer;
272  totsti : array (our_integer range 1..ntot) of our_integer;
273
274  -- variables that are shared between tasks in microsecond level tasking tests
275  -- this occurs in tests where mstop is called in a different task than
276  -- tinit, tstart, tstopm and tprint
277
278  pragma shared(micind); --X
279  pragma shared(strmic); --X

```

Table D-2. OURSYS Package Body Declarations (Continued)

```

280
281      pragma interface (FORTRAN, gettim);  --X
282      pragma interface (FORTRAN, gstats);  --X
283      pragma interface (FORTRAN, initim);  --X
284      pragma interface (FORTRAN, istats);  --X
285      pragma interface (FORTRAN, secs);    --X
286
287
288      -- XXXXX oursys procedures follow in alphabetical order XXXXX --
289
290      -- XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX --
291

```


Table D-3 OURSPC Package Specification

```

1  -- Ada Real time/Run time Environment Test      Aerospace Corporation  --
2  --                                              --
3  -- Test:      .....                          --
4  -- File:      OURSPC                          --
5  --                                              --
6  --          REK          2/1/88                --
7  --                                              --
8  -- Description:                                --
9  --                                              --
10 -- **** This package defines datatypes and objects used in tests for --
11 -- **** access/enumeration/record types, representation specifications, --
12 -- **** unchecked programming, shared variables and tasking features. --
13 -- **** It also defines procedures which set/reference the objects --
14 -- **** defined --
15 -- ****
16 -- **** Note: In the Ada version of OURSPC, statements followed by --X
17 -- **** are implementation dependent and may need to be changed --
18 -- **** for each compiler and for each target machine. --
19 -- **** Changes may also be required to the corresponding --
20 -- **** statements in the JOVIAL and FORTRAN versions of OURSPC. --
21 -- ****
22 -- ****
23 with system;
24 with unchecked_conversion;
25 with calendar; use calendar;
26 with oursys; use oursys;
27 package ourspc is
28 --
29 -- ****
30 -- data types used in tests --
31 -- ****
32 --
33     type packed_boolean is new boolean;
34     type packed_bool_array is new boolean_vector (1..10);
35
36     type months is (Jan, Feb, Mar, Apr, May, Jun,
37                   Jul, Aug, Sep, Oct, Nov, Dec);
38     type rep_months is new months;
39
40     type five_bit_integer is range 0..31;
41     type packed_months is new months;
42
43     type records is
44         record
45             comp_i1 : five_bit_integer;
46             comp_i2 : five_bit_integer;
47             comp_b1 : packed_boolean;
48             comp_b2 : packed_boolean;
49             comp_e : packed_months;
50         end record;
51     type packed_records is new records;
52     type rep_records is new records;
53
54     type variant_records(discr : months := Jun) is
55         record
56             case discr is
57                 when jan =>
58                     first : our_integer;
59                 when mar =>
60                     second : our_float;
61                 when others =>
62                     comp_i1 : five_bit_integer;
63                     comp_i2 : five_bit_integer;
64                     comp_b1 : packed_boolean;
65                     comp_b2 : packed_boolean;
66                     comp_e : packed_months;
67             end case;

```

Table D-3 OURSPC Package Specification (continued)

```

68         end record;
69     type packed_variant is new variant_records;
70
71     type records_array is array(our_integer range 1..10) of records;
72     type packed_records_array is array(our_integer range 1..10)
73         of packed_records;
74     type rep_records_array is array(our_integer range 1..10) of rep_records;
75
76     type Enumeration is (Ident_1,
77                         Ident_2,
78                         Ident_3,
79                         Ident_4,
80                         Ident_5);
81
82     subtype One_To_Fifty is our_integer range 1 .. 50;
83
84     type String_10 is array(our_integer range 1..10) of character;
85     type String_20 is array(our_integer range 1..20) of character;
86     type String_30 is array(our_integer range 1..30) of character;
87
88     type Record_Type;
89
90     type Record_Pointer is access Record_Type;
91
92     type Record_Type is
93     record
94         Pointer_Comp : Record_Pointer;
95         Enum_Comp    : Enumeration;
96         Int_Comp     : One_To_Fifty;
97         String_Comp  : String_30;
98     end record;
99
100    -- ***** --
101    -- representation specifications for declared types(placed before object --
102    -- declarations) --
103
104    pragma pack(packed_bool_array); --X
105    for packed_boolean use (0,1); --X
106    for packed_boolean'size use 1; --X
107
108    for rep_months use (1,2,4,8,16,32,64,128,
109                      256,512,1024,2048); --X
110    for rep_months'size use 12; --X
111
112    pragma pack(packed_records); --X
113    pragma pack(packed_variant); --X
114    pragma pack(String_10); --X
115    pragma pack(String_20); --X
116    pragma pack(String_30); --X
117
118    for five_bit_integer'size use 5; --X
119    for packed_months'size use 4; --X
120
121    -- define rep-records to be sixteen bits in size --
122    -- align rep-records on a 16 bit boundary --
123    for rep_records use
124    record at mod (16 / system.storage_unit); --X
125        comp_i1 at 0 range 0..4; --X
126        comp_i2 at 0 range 5..9; --X
127        comp_b1 at 0 range 10..10; --X
128        comp_b2 at 0 range 11..11; --X
129        comp_e at 0 range 12..15; --X
130    end record;
131
132    pragma pack(packed_records_array); --X
133    for rep_records_array'size use 10 * 16; --X
134    --

```

Table D-3 OURSPC Package Specification (continued)

```

135 -- ***** --
136 -- unchecked conversions --
137 -- ***** --
138
139     function float_to_integer is new unchecked_conversion(
140         our_float,our_integer); --X
141     function records_to_integer is new unchecked_conversion(
142         rep_records,our_integer); --X
143 --
144 -- ***** --
145 -- test constants --
146 -- ***** --
147
148 -- aggregate constants used in array,record tests and to initialize --
149 -- array,record variables --
150
151     string_30_con : constant String_30 := "123456789012345678901234567890";
152
153     record_type_con : constant record_type := (null,Ident_1,50,
154         string_30_con);
155     records_con_1 : constant records := (1,1,false,true,Jan);
156     records_con_2 : constant records := (2,2,false,false,Feb);
157     records_con_3 : constant records := (3,3,true,false,Mar);
158     pack_records_con_1 : constant packed_records := (2,2,false,false,Feb);
159     pack_records_con_2 : constant packed_records := (1,1,false,true,Jan);
160     pack_records_con_3 : constant packed_records := (3,3,true,false,Mar);
161     rep_records_con_1 : constant rep_records := (3,3,true,false,Mar);
162     rep_records_con_2 : constant rep_records := (2,2,false,false,Feb);
163     rep_records_con_3 : constant rep_records := (1,1,false,true,Jan);
164     variant_con_1 : constant variant_records := (Jun,1,1,false,true,Jan);
165     variant_con_2 : constant variant_records := (Jun,2,2,false,false,Feb);
166     pack_variant_con_1 : constant packed_variant := (Jun,2,2,false,
167         false,Feb);
168     pack_variant_con_2 : constant packed_variant := (Jun,1,1,false,
169         True,Jan);
170     bool_array_con_1 : constant boolean_array_10 := (1..10 => true);
171     bool_array_con_2 : constant boolean_array_10 := (1..10 => false);
172     pack_bool_con_1 : constant packed_bool_array := (1..10 => true);
173     pack_bool_con_2 : constant packed_bool_array := (1..10 => false);
174
175 -- max_time - delay time used in MASTER task timed entry calls. It
176 -- contains the maximum time that a task test should take
177 -- in seconds.
178     max_time : constant duration := 60.0; --X
179
180 -- time_slice_interval - the forced time slice interval used in tasking
181 -- tests. It should be consistent across Ada
182 -- run-time environments for each target machine
183 -- (for the VAX = .01secs)
184     time_slice_interval : constant duration := 0.01; --X
185
186 -- driver_priority - priority of driver tasks in tasking tests --
187 -- master_priority - priority of master tasks in tasking tests --
188 -- slave_priority - priority of slave or master created tasks in
189 -- tasking tests --
190
191     driver_priority : constant our_integer := 3; --X
192     master_priority : constant our_integer := 2; --X
193     slave_priority : constant our_integer := 1; --X
194
195 -- ***** --
196 -- procedure definitions used by oursys --
197 -- ***** --
198
199 -- spcset - sets all ourspc.shrX shared global variables to oursys.tmpX
200 -- equivalents and also sets ourspc.tmpX variables.
201

```

Table D-3 OURSPC Package Specification (continued)

```

202  -- spcref - references all ourspc.shr*,tmp* global variables
203
204      procedure spcset;
205
206      procedure spcref;
207
208
209  -- ***** --
210  -- global variable definitions --
211  -- ***** --
212
213  -- non-shared record structure variables
214
215      tmper0,tmper1,tmper2,tmper3,tmper4,tmper5,
216      tmper6,tmper7,tmper8,tmper9,tmpera      : rep_months;
217      tmpes0,tmpes1,tmpes2,tmpes3,tmpes4,tmpes5,
218      tmpes6,tmpes7,tmpes8,tmpes9,tmpesa      : months;
219      tmppp0,tmppp1,tmppp2,tmppp3,tmppp4,tmppp5,
220      tmppp6,tmppp7,tmppp8,tmppp9,tmpppa      : record_pointer;
221      tmppr0,tmppr1,tmppr2,tmppr3,tmppr4,tmppr5,
222      tmppr6,tmppr7,tmppr8,tmppr9,tmppra      : record_type;
223      tmprc0,tmprc1,tmprc2,tmprc3,tmprc4,tmprc5,
224      tmprc6,tmprc7,tmprc8,tmprc9,tmprca      : records;
225      tmprp0,tmprp1,tmprp2,tmprp3,tmprp4,tmprp5,
226      tmprp6,tmprp7,tmprp8,tmprp9,tmprpa      : packed_records;
227      tmprr0,tmprr1,tmprr2,tmprr3,tmprr4,tmprr5,
228      tmprr6,tmprr7,tmprr8,tmprr9,tmprra      : rep_records;
229      tmpvs0,tmpvs1,tmpvs2,tmpvs3,tmpvs4,tmpvs5,
230      tmpvs6,tmpvs7,tmpvs8,tmpvs9,tmpvsa      : variant_records;
231      tmpvp0,tmpvp1,tmpvp2,tmpvp3,tmpvp4,tmpvp5,
232      tmpvp6,tmpvp7,tmpvp8,tmpvp9,tmpvpa      : packed_variant;
233
234  -- non-shared arrays --
235
236      tmcp1,tmcp4,tmcp5 : String_10;
237      tmcp2 : String_20;
238      tmcp3 : String_30;
239
240      tmpras : records_array;
241      tmprap : packed_records_array;
242      tmprar : rep_records_array;
243
244      tmppb1,tmppb2,tmppb3 : packed_bool_array;
245
246  -- shared integer variables ( all variables = 3 except shris0 which is 0) --
247  -- set/used --
248
249  -- task test work counter --
250      total_work : our_integer;
251      pragma shared(total_work); --X
252      shris0 : our_integer;
253      shris1 : our_integer;
254      shris2 : our_integer;
255      shris3 : our_integer;
256      shris4 : our_integer;
257      shris5 : our_integer;
258      shris6 : our_integer;
259      shris7 : our_integer;
260      shris8 : our_integer;
261      shris9 : our_integer;
262      shrisa : our_integer;
263      shrisb : our_integer;
264      shrisc : our_integer;
265      shrisd : our_integer;
266      shrise : our_integer;
267      pragma shared(shris0); --X
268      pragma shared(shris1); --X

```

Table D-3 OURSPC Package Specification (concluded)

```

269      pragma shared(shris2);  --X
270      pragma shared(shris3);  --X
271      pragma shared(shris4);  --X
272      pragma shared(shris5);  --X
273      pragma shared(shris6);  --X
274      pragma shared(shris7);  --X
275      pragma shared(shris8);  --X
276      pragma shared(shris9);  --X
277      pragma shared(shrisa);  --X
278      pragma shared(shrisb);  --X
279      pragma shared(shrisc);  --X
280      pragma shared(shrisd);  --X
281      pragma shared(shrise);  --X
282
283      -- variables of type time and duration --
284
285      time_0,time_1,time_2,time_3,time_4,time_5    : time;
286      time_6,time_7,time_8,time_9,time_a           : time;
287      duration_0,duration_1,duration_2,duration_3  : duration;
288      duration_4,duration_5,duration_6,duration_7  : duration;
289      duration_8,duration_9,duration_a              : duration;
290  end ourspsc;

```

Table D-4 OURTYP Package Specification

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --                                              --
3  -- Test: .....                                --
4  -- File:   OURTYP                                --
5  --                                              --
6  --       ARA,REX   2/1/88                                --
7  --                                              --
8  -- Description:                                --
9  --                                              --
10 -- XXXX This package declares compiler specific non-standard integer --
11 -- XXXX and floating point data types . It --
12 -- XXXX also declares global variables with these datatypes that are --
13 -- XXXX used by relevant benchmark programs. --
14 --                                              --
15 with system; use system;
16 with oursys; use oursys;
17 package ourtyp is
18
19     subtype our_short_integer      is short_integer      ;--X
20     subtype our_short_short_integer is short_short_integer ;--X
21     subtype our_long_float         is long_float         ;--X
22     subtype our_long_long_float    is long_long_float    ;--X
23
24 -- typset      - initialize the non-32 bit standard variables --
25 --                                              --
26 -- typref      - reference the non-32 bit standard variables --
27 --                                              --
28     procedure typset;
29
30     procedure typref;
31
32
33     tsi0   : our_short_integer;
34     tsi1   : our_short_integer;
35     tsi2   : our_short_integer;
36     tsi3   : our_short_integer;
37     tsi4   : our_short_integer;
38     tsi5   : our_short_integer;
39     tsi6   : our_short_integer;
40     tsi7   : our_short_integer;
41     tsi8   : our_short_integer;
42     tsi9   : our_short_integer;
43     tsia   : our_short_integer;
44     type s_int_ary_10 is array(our_integer range 1..10)
45         of our_short_integer;
46     type s_int_ary_10_10 is array(our_integer range 1..10,
47         our_integer range 1..10) of our_short_integer;
48     pragma pack(s_int_ary_10); --X
49     pragma pack(s_int_ary_10_10); --X
50     tsia1,tsia2,tsia3 : s_int_ary_10;
51     tsia4 : s_int_ary_10_10;
52
53     tssi0   : our_short_short_integer;
54     tssi1   : our_short_short_integer;
55     tssi2   : our_short_short_integer;
56     tssi3   : our_short_short_integer;
57     tssi4   : our_short_short_integer;
58     tssi5   : our_short_short_integer;
59     tssi6   : our_short_short_integer;
60     tssi7   : our_short_short_integer;
61     tssi8   : our_short_short_integer;
62     tssi9   : our_short_short_integer;
63     tssia   : our_short_short_integer;
64     type s_s_int_ary_10 is array(our_integer range 1..10)
65         of our_short_short_integer;
66     type s_s_int_ary_10_10 is array(our_integer range 1..10,
67         our_integer range 1..10) of our_short_short_integer;

```

Table D-4 DURTYP Package Specification (concluded)

```

68      pragma pack(s_s_int_ary_10_10); --x
69      pragma pack(s_s_int_ary_10); --x
70      tssia1,tssia2,tssia3 : s_s_int_ary_10;
71      tssia4 : s_s_int_ary_10_10;
72
73      tlf0    : our_long_float;
74      tlf1    : our_long_float;
75      tlf2    : our_long_float;
76      tlf3    : our_long_float;
77      tlf4    : our_long_float;
78      tlf5    : our_long_float;
79      tlf6    : our_long_float;
80      tlf7    : our_long_float;
81      tlf8    : our_long_float;
82      tlf9    : our_long_float;
83      tlfa    : our_long_float;
84      type lflt_ary_10 is array(our_integer range 1..10)
85          of our_long_float;
86      type lflt_ary_10_10 is array(our_integer range 1..10,
87          our_integer range 1..10) of our_long_float;
88      tlfa1,tlfa2,tlfa3 : lflt_ary_10;
89      tlfa4 : lflt_ary_10_10;
90
91      tllf0    : our_long_long_float;
92      tllf1    : our_long_long_float;
93      tllf2    : our_long_long_float;
94      tllf3    : our_long_long_float;
95      tllf4    : our_long_long_float;
96      tllf5    : our_long_long_float;
97      tllf6    : our_long_long_float;
98      tllf7    : our_long_long_float;
99      tllf8    : our_long_long_float;
100     tllf9    : our_long_long_float;
101     tllfa    : our_long_long_float;
102     type l_lflt_ary_10 is array(our_integer range 1..10)
103         of our_long_long_float;
104     type l_lflt_ary_10_10 is array(our_integer range 1..10,
105         our_integer range 1..10) of our_long_long_float;
106     tllfa1,tllfa2,tllfa3 : l_lflt_ary_10;
107     tllfa4 : l_lflt_ary_10_10;
108
109     end ourtyp;
110

```

Table D-5 OURDMP Package Specification

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --                                              --
3  -- Test:      .....                          --
4  -- File:      OURDMP                          --
5  --                                              --
6  --          REK    2/1/88                      --
7  --                                              --
8  -- Description:                                --
9  --                                              --
10 -- **** This package defines procedures used by test programs to --
11 -- **** dump the contents of various global variables defined by OURSYS--
12 -- **** If global variable DEBUG is OFF, then no print is generated --
13
14 with OURSYS; use OURSYS;
15 with OURSPC; use OURSPC;
16 package OURDMP is
17 --
18 --
19 -- dumpb(num,a1,a2,a3,a4,a5) - print a record of its boolean arguments --
20 --                               - num specifies the number of arguments to print --
21 --
22 -- dumpi(num,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10) -
23 --                               - print a record of its integer arguments --
24 --                               - num specifies the number of arguments to print --
25 --
26 -- dumpr(num,a1,a2,a3,a4,a5) - print a record of its real arguments --
27 --                               - num specifies the number of arguments to print --
28 --
29 -- dumpt                               - print global test variables tmpX --
30 --
31 procedure dumpb(num : in our_integer;a1,a2,a3,a4,a5 : in boolean);
32 procedure dumpi(num,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10 : in our_integer);
33 procedure dumpr(num : in our_integer;a1,a2,a3,a4,a5 : in our_float);
34 procedure dumpt;
35 end OURDMP;

```


Table D-6 MATHFUN Package Specification

```

1  -- Ada Real Time/Run Time Environment Test      Aerospace Corporation --
2  --
3  -- Test:          .....
4  -- File:          MATHFUN
5  --
6  --          REK    2/1/88
7  --
8  -- Description:
9  --
10 -- **** This file makes standard math functions used by the WHETSTONE
11 -- **** benchmark available to test programs. This is done by
12 -- **** instantiation of generic program libraries with the type
13 -- **** our_float. The names of any library routines which are different
14 -- **** from those used in the WHETSTONE benchmark (sin,cos,arctan,sqrt,
15 -- **** exp,ln) should be renamed
16
17 with math_lib;
18 WITH OURSYS; USE OURSYS;
19 PACKAGE MATHFUN IS
20     package ourfloat_lib is new math_lib(our_float); --x
21     use ourfloat_lib;
22     function arctan(x : our_float) return our_float renames atan; --x
23     function      ln(x : our_float) return our_float renames log; --x
24 END MATHFUN;

```

APPENDIX E

ACPS TEST COMPARISON TOOL

The ACPS test comparison tool automatically compares ACPS compile-time and run-time test statistic files and generates formatted outputs of the input files and of the comparison results. The tool consists of two programs which compare compile-time and run-time test results. The programs were developed using a VAX/VMS host and were designed to process ACPS test result output specific to VAX/VMS. Appendix C describes the VAX/VMS command procedures used to execute the tool. The programs are written in FORTRAN 77 and use the VAX FORTRAN INCLUDE statement for reference to FORTRAN common block definitions. The input/output files are read/written with FORTRAN formatted read/write statements. The target operating system dependencies are separated into separate common block include files and subroutines. The input/output logical unit assignments for the tools are shown in Tables E-1 and E-2.

This appendix describes the format and target operating system dependencies of all comparison tool input and output files. It also describes the required order in which test records must appear within the input test statistic files. The discussion is illustrated with sample input and output files generated from execution of the VAX/VMS version of the ACPS using command procedures described in Appendix C.

E.1 ACPS COMPILE-TIME TEST RESULT COMPARATOR (CCOMP)

E.1.1 Compile-Time Test Result Input File Formats

CCOMP takes as input two files containing compile-time generated test records augmented by compiler and linker diagnostic and informational messages. Table E-3 shows output generated using DEC VAX Ada compiler and Table E-4 shows output generated using the ECSPO JOVIAL compiler. The output contains two types of records that are processed by the comparison tool:

- a. operating system-independent test records which begin with the symbol < in column 1
- b. operating system-dependent statistic records for VAX/VMS, which are generated by the DCL SHOW STATUS command

The output is divided into the following five sections: compilation test records, linkage edit test records, source module size test records, object module size test records, and executable module test records. These sections are formatted as follows:

a. Compilation Test Record Section Format

The compilation test record section contains records in the following order:

```
<start compiles
<FILE-NAME
    DCL SHOW STATUS RECORD 1
    DCL SHOW STATUS RECORD 2
    o
    o
    o
<FILE-NAME
    DCL SHOW STATUS RECORD 1
    DCL SHOW STATUS RECORD 2
<end compiles
```

where:

FILE-NAME - is either the name "system" or an ACPS file name whose syntax is described in Section 6.1. The name "system" is used to represent the compilation of the language-specific test support software.

b. Linkage Edit Test Record Section Format

The linkage edit test record section format consists of records in the following order:

```
<start links
<FILE-NAME
    DCL SHOW STATUS RECORD 1
    DCL SHOW STATUS RECORD 2
    o
    o
    o
<FILE-NAME
    DCL SHOW STATUS RECORD 1
    DCL SHOW STATUS RECORD 2
<end links
```

where:

FILE-NAME - is an ACPS filename whose syntax is described in Section 6.1.

c. Source Module Size Test Record Section Format

The source module size test record section contains records in the following order:

```
<start source size
<FILE-NAME #
    o
    o
<FILE-NAME #
<end source size = ###
```

where:

FILE-NAME - is either the name "system" or an ACPS filename whose syntax is in Section 6.1. The name "system" is used for all source files in the language-specific test support software.

- is the source module size. For VAX/VMS, it is expressed in 512 byte disk blocks.

- is the total size of source modules. For VAX/VMS, it is expressed in 512 byte blocks.

d. Object Module Size Test Record Section Format

The object module size test record section contains records in the following order:

```
<start object size
<FILE-NAME #
    o
    o
<FILE-NAME #
<end object size = ###
```

where:

FILE-NAME - is either the name "system" or an ACPS filename whose syntax is in Section 6.1. The name "system" is used for all object files in the language-specific test support software.

- is the object module size. For VAX/VMS, it is expressed in 512 byte disk blocks

- is the total size of object modules for VAX/VMS, it is expressed in 512 byte blocks.

e. Executable Module Size Test Record Section Format

The executable module size test record section contains records in the following order:

```
<start executable size
<FILE-NAME 1 #
    o
    o
<FILE-NAME N #
<end executable size = ###
```

where:

FILE-NAME - is an ACPS filename whose syntax is section in Section 6.1.

- is the executable module size. For VAX/VMS, it is expressed in 512 byte disk blocks

- is the total size of executable modules. For VAX/VMS, it is expressed in 512 byte blocks.

E.1.2 User Input

CCOMP interacts with the user by requesting user responses to the following questions:

- a. What title should be placed on the comparison output listings?
- b. Which host operating system was used to generate the input files?
The tool currently only supports output generated by VAX/VMS.

E.1.3 CCOMP Output File Formats

CCOMP generates two types of output files: one containing formatted versions of the input files and one containing the results of the comparison of the input files. The formats of these files are as follows:

a. Formatted Input File Format

Figures E-3 and E-4 contain formatted output files generated by CCOMP from the VAX Ada and ECSP0 JOVIAL compilation statistics shown in Figures E-1 and E-2. As can be seen from these figures, the CCOMP outputs for different test languages have the same content and format. The outputs are divided into five sections: compilation statistics, linkage edit statistics, source module size statistics, object module size statistics, and executable or load module size statistics. The heading for each output shows the logical unit number (e.g., UNIT 8 in Figure E-3) of the input file from which the output is derived.

The compilation and linkage edit outputs have identical formats. The outputs are in tabular form. The separate entry "totals" is used to contain the summation by column of the values of all other entries in the table. These outputs are operating system-dependent since they contain statistics generated by the DCL SHOW STATUS command. The meanings of the output headings are as follows:

FILE(S) - the name of a single file or group of files. The name "system" corresponds to the files associated with the language-specific test support software

WALL-CLOCK - the elapsed time expressed in units of seconds

CPU-TIME - the VMS CPU time statistics expressed in units of seconds

BUFIO - the number of VMS buffered input/output operations

DIRIO - the number of VMS direct input/output operations

PAGE-FAULTS - the VMS working set page fault statistic

The source, object, and executable module size statistics section have identical format and contain data which is operating system-independent (albeit obtained from operating system-dependent commands). The file sizes are expressed in units of 512 byte disk blocks.

b. Comparison Result File Format

The comparison result file contains output that compares the compile-time statistics from the two input files supplied to CCOMP. Table E-7 shows an example comparison output file for the input files shown in Tables E-3 and E-4. The output comparison results are presented in fractional form as a ratio of the test statistic from input logical unit 8 divided by the corresponding test statistic from input logical unit 7. The file names appearing on the output are the same as those appearing on logical unit 8. Test statistics that appear on one input file but not on the other are not compared and do not affect the comparison output. The output is divided into six sections: the user/tool dialogue, the compilation comparison, the linkage edit comparisons, the source module size comparison, the object module size comparison, and the load or executable module size comparison.

The first page of Table E-7 shows the dialogue between the user and CCOMP. User responses follow the record containing the prompt character < while CCOMP requests precede the prompt record.

The comparison output sections are in the same format as the corresponding sections in the formatted input file outputs described above, the only difference being that the tabular data contains fractional comparisons between corresponding input file statistics. For example, in the compilation comparison output, the total compile time was 5% less in Ada than in JOVIAL even though the number of page faults generated was 1.8 times greater in Ada than in JOVIAL.

E.2 ACPS RUN-TIME RESULT COMPARATOR (CEXEC)

E.2.1 Run-Time Test Result Input File Formats

CEXEC takes as input two files containing run-time generated test records optionally augmented by target operating system-dependent records that measure the program loading time as well as the program execution time. Tables E-8 and E-9 show sample output generated by run-time execution of ACPS tests compiled by the VAX Ada compiler and the ECSP0 JOVIAL compiler, respectively. As can be seen from these outputs, the run-time statistic files generated by different ACPS test languages have identical formats. The run-time output file contains two types of records that are processed by CEXEC:

- a. Optional target operating system-dependent records. These are optionally processed based on user response to questions posed by CEXEC. Their intent is to permit measurement of the program loading time of the test overhead programs and of the entire test suite.

Three records are generated by the ACPS VAX/VMS run-time execution command procedure using the DCL SHOW STATUS command. The records appear immediately before and after the overhead test record and also after the last test record as shown in Table E-8.

- b. Test support-software test records. These are generated by the language-specific test support software and may contain operating system-dependent statistic fields. They begin with the symbol < preceded by either one or two blanks (dependent upon how Ada compilers generate output to the standard output file. For both FORTRAN and JOVIAL, the < symbol is preceded by a single blank).

Each test record contains two logical records. The first logical record contains the test name, the elapsed time, and operating system dependent records as follows:

<u>Column</u>	<u>Contents</u>
2	<
3-9	Test name (described in Section 6.2).
10-19	Elapsed time expressed in units of seconds to microsecond level accuracy.
20-29	Number of test iterations (an integer). The field is zero for microsecond level tests.

30-119 Nine target operating system-dependent statistics expressed as integers. For VAX/VMS, the statistics appear in the following order:

1. CPU time expressed in units of 10 ms
2. number of page faults
3. number of buffered input/output operations
4. number of direct input/output operations
5. number of open files
6. P0 virtual address space used expressed in bytes
7. P1 virtual address space used expressed in bytes
8. process page count
9. global page count

The second logical record contains data dependent on the type of the test. The record contains up to six 10-column fields. The first field always specifies the number of data fields that follow. For nonmicrosecond level tests (e.g., AA00000), the first data field always contains the size of the compiled test code in bytes. Subsequent data fields contain test-specific statistics that are only defined in the test source code prologue. If a data field contains a -1, CEEXEC ignores the field since the statistic represented involves a language feature that is not supported by the corresponding compiler (e.g., VAX Ada run-time tests do not include the size of the compiled test code, since the VAX Ada compiler does not support the address attribute for labels). For microsecond level tests (e.g., AAM0000), the data records correspond to the maximum and median elapsed time measurements made and are expressed in units of microseconds.

There are five types of test records: informational tests, microsecond level tests, self-measured overhead tests; optimization comparison tests, single iteration tests, and multiple iteration tests. Tables E-8 and E-9 contain examples of each type of test record.

The informational test records are identified by name (i.e., I in character position three) and contain informational rather than execution performance statistics. The data in these records is used to describe the size of data objects and data types, and the distribution of work in task loading tests. The data is entirely contained in the second logical record. Records of this type are not compared by CEEXEC but are written to the corresponding formatted output file.

The microsecond level tests are identified by test name (i.e., M or N in character position three) and contain zero in the test iteration field.

The self-measured overhead test records are identified by name (i.e., S in character position three) and are records in which the test overhead has been subtracted from every statistic in the first and second logical records. The test record appearing immediately before the self-measured test record corresponds to the same test with the test overhead included in the statistics.

The optimization comparison test record is also identified by name (i.e., C in character position three) and corresponds to the comparison of hand-optimized tests to compiler-optimized tests. All statistics are fractional and are computed as a ratio of 100 times the compiler-optimized test statistics divided by the corresponding hand-optimized test statistics. Values of 100 correspond to statistics that are the same (see the record for A0C0508 in Figure E-6 which compares A000508 to A000507).

The single iteration test record has a one in the test iteration field of the first logical record (columns 20-29). It corresponds to execution of the first iteration through the test loop. The multiple iteration test record is used for the remaining iterations of the test loop. For this type of test record, the second logical record is blank and contains no data fields. All code and data object sizes for the multiple iteration tests appear in the second logical record of the corresponding single-iteration test. (e.g., see test JA00000 in Table E-9.

As shown in Table E-8, the run-time statistic files input to CEXEC also contain diagnostic and informational messages generated by the test support software. These messages are preceded by four asterisks (*) and are output by the test support procedure tfail. The messages that can be generated are as follows:

**** Elapsed time in above test is too short (< 100 times clock granularity).

This message occurs if test durations are not long enough to ensure that the error in elapsed time measurements is less than 1%.

For corrective action, the user can increase the corresponding test iteration count.

**** Test <test-name> fails - incorrect value computed.

This message occurs in self-monitoring tests that compare computed values against expected values.

For corrective action, the user should examine the test source and compiled code to determine the cause of the error.

**** Test <test-name> fails - incorrect path taken.

This message occurs in self-monitoring tests when a test statement is unexpectedly executed.

For corrective action, the user should examine the test source and compiled code to determine the cause of the error.

**** Test <test-name> fails - error in reading operating system statistics.

This message occurs in VAX/VMS test support software whenever the system service SYS\$GETJPI returns an error condition.

This message should not be generated and signals that a fatal error has occurred in the test support software.

**** Test <test-name> fails - comparison tests are improperly used.

For corrective action, the user should make sure that the test name conforms to the test naming conventions described in Section 6 and that the test support software was also used properly. Optimization comparison tests are discussed in Section 4.3.

**** Test <test-name> fails - difference tests should have the same number of iterations as the loop overhead test.

This message occurs when test names or self-measuring overhead tests are improperly used.

For corrective action, the user should make sure the test name follows the test naming convention described in Section 6 and that the test coding conventions described in Section 4 are followed.

**** Test <test-name> fails - pragma priority is not supported with enough distinct priority levels.

This message occurs in tasking tests that require three different task priority levels to work properly.

No corrective action can be taken.

**** Test <test-name> fails - task test does not complete within ourspc.max_time seconds.

This message is generated in task loading test drivers which assume that the test can complete within a specified time (60 sec for DEC VAX).

For corrective action increase ourspc.max_time. If this does not correct the problem, then the run-time environment probably does not support preemptive task scheduling and no further corrective action is recommended.

**** Test <test-name> fails - change of representation between objects does not preserve the values.

This message is generated in self-monitoring tests which check the results of operations which change representation between record objects.

For corrective action, the user should examine the source and compiled code to determine the cause of the error.

**** Test <test-name> fails - unchecked conversion incorrectly converted a floating point value (3.0) to an integer value (3).

For corrective action, examine the source and compiled code to determine the cause of the error.

**** Test <test-name> fails - constraint error

**** Test <test-name> fails - numeric error

**** Test <test-name> fails - program error

**** Test <test-name> fails - storage-error

**** Test <test-name> fails - tasking-error

**** Test <test-name> fails - nonpredefined exception

The above messages are generated in Ada type E tests that fail as stated in the message.

For corrective action, the user should examine the test source code for the cause of the error.

All of the above messages are ignored by CEXEC.

E.2.2 User Input

CEXEC is an interactive program that requests information from the user to direct processing of the input files and to affect how the output files are generated. CEXEC requests user responses to supply the following information:

- a. Whether or not to generate a maximum or minimum test result file.
- b. Whether or not to process operating system status records. These records should not be processed in executions involving maximum or minimum test result files.
- c. Whether or not to process operating system-dependent statistics in test records.

- d. Which target operating system was used. This request is only made if operating system-dependent data are to be processed. Currently, CEXEC only supports the VMS operating system.
- e. Whether or not to subtract the operating statistics obtained from the test overhead record from corresponding statistics in subsequent tests.
- f. The title to be placed on all output files generated by CEXEC.
- g. The minimum test execution time. CEXEC generates an error message in the formatted output file for multiple iteration tests if the test execution time exclusive of test loop overhead is less than the minimum time specified by the user.

The first page of Tables E-12 and E-15 show sample dialogues between the user and CEXEC. The CEXEC prompt record contains a single < symbol. CEXEC requests for information immediately precede the prompt record while user responses follow the prompt record.

E.2.3 CEXEC Output File Formats

CEXEC generates two types of output files: one containing formatted versions of the input files and one containing the results of the input file comparisons. There are two formats for each type of output: one with and one without target operating system dependent information. The formats of these two types of output files are as follows:

- a. Formatted input file format. Tables E-10 and E-11 show formatted input files generated by CEXEC from the VAX Ada and ECSPO JOVIAL input files shown in Tables E-8 and E-9. The outputs shown contain only operating system-independent statistics and are generated by compiler-independent source code within the language-specific test support software. As can be seen from these tables, E-10 and E-11, the format and content of these outputs are the same for each test language. The listing heading shows the test language associated with the output and the logical unit numbers [e.g., unit (8) in Table E-10] of the input file from which the output is derived. The output is presented in tabular form with one entry per test record. An "elapsed time too short" error message can appear in the output whenever the elapsed time in the immediately preceding multiple iteration test is less than a user specified minimum value. The meanings of the column headings are as follows:

- TEST-ID - Contains the name of the test whose syntax is described in Section 6.2.
- WALL-TIME - Contains the elapsed test execution time in seconds. The effect of the test loop overhead (tests AA00000, AAM0000) has been removed from all tests except the optimization comparison (e.g., AOC0508) and self-measuring overhead (e.g., AFS5307) tests.
- ITERATIONS - Contains the number of test loop iterations executed.
- RUN STATISTICS - Contains five entries corresponding to the data fields appearing in the second logical record of each test input record. An entry containing 0* indicates that the data field does not apply to this test. The effect of the test loop overhead has been removed for microsecond level tests (e.g., AGM0008) and for single-iteration tests.

Tables E-13 and E-14 show formatted input files generated by CEEXEC that also contain VAX/VMS operating system statistics. Each output test record contains two logical records: the first record contains nonoperating dependent data and the second record contains operating system-specific data. The effect of the test loop overhead is optionally subtracted from the operating system-dependent statistics. The meanings of the column headings for VAX/VMS statistics are as follows:

- CPUTIM - the VMS CPU time statistic expressed in units of 10 ms
- PAGEFLT - the number of working set page faults
- BUFIO - the number of buffered input/output operations
- DIRIO - the number of direct input/output operations
- FILCNT - the number of open input/output files
- POSPACE - the amount of P0 virtual address space used
- PISPACE - the amount of P1 virtual address space used
- PPGCNT - the process page count
- GPGCNT - the global page count

b. Comparison result file format. The comparison result file contains output that compares the run-time statistics contained on the two input files (e.g., Tables E-8 and E-9) read by CEXEC. Test statistics that appear on one input file but not on another are not compared and do not affect the comparison output. The output comparison results are presented in fractional form as the ratios of the test statistics from input on logical unit 8 divided by test statistics from input on logical unit 7. The effect of the number of iterations is taken into account in both the elapsed time and CPU time (if present) statistics. The test names appearing on the outputs are the same as those contained on logical unit 8.

Table E-15 shows a comparison output file that does not contain target operating system-specific data. The first page shows the tool/user dialogue which causes suppression of processing of operating system-dependent records and fields. The meanings of the output headings are the same as for the formatted input file output shown in Table E-10 and described previously. Test comparison totals, which appear at the end of the output, present accumulated comparisons of all but the test overhead tests and the optimization comparison tests. The meanings of the entries appearing in the test total output are as follows:

micro	-	microsecond level test totals
1 iter	-	single-iteration test totals
>1 iter	-	multiple-iteration test totals (not including self-measuring overhead tests)
diff.	-	self-measuring overhead test totals

Table E-15 shows a comparison output file that contains VMS status record and VMS statistic field comparisons. The first page of the output shows the dialogue between the tool and user that selected the form of output. The rest of the output is divided into three sections: benchmark test comparisons, SHOW STATUS statistics, and SHOW STATUS comparisons.

The benchmark test comparison section contains the individual test comparisons. The meanings of the output headings are the same as for the formatted input file output shown in Table E-13 and described above. The results are expressed as the ratio of the processed statistics from input logical unit 8 divided by the corresponding processed statistics from input logical unit 7. The values of the processed statistics used are contained in the formatted input file outputs shown in Tables E-13 and E-14. Test comparison totals appear at the end of the output. The entries have the same meaning as those in Table E-12 which were previously described.

The VMS SHOW STATUS statistics section displays the result of processing the VMS SHOW STATUS records that are inserted into the run-time data file to measure both the program loading and test execution time of the overhead test and of the entire test suite. The meanings of the output headings are as follows:

- I/O-UNIT - input logical unit containing the data
- NUMBER - specifies the pair of status records processed as follows:
 - a. corresponds to the first pair which measure the execution and program loading time of the overhead test
 - b. corresponds to the second pair which measure the execution and program loading time of all other tests executed
- WALL-CLOCK - elapsed time in seconds
- CPU-TIME - VMS CPU time statistic in seconds
- BUFIO - number of buffered input/output operations
- DIRIO - number of direct input/output operations
- PAGE-FAULTS - number of working set page faults

The VMS SHOW STATUS comparison section compares the results contained in the SHOW STATUS statistics section for corresponding statistic records contained on input logical units 8 and 7. The comparison results are expressed as the ratios of the test statistics from logical unit 8 divided by the corresponding test statistics on logical unit 7. The meanings of the output column headings are the same as those in the SHOW STATUS statistics section previously described.

E.2.4 Measurement of Test Repeatability

For target machines for which the run-time environment cannot be constrained to permit repeatable measurements of test execution times, CEEXEC can be used to measure the repeatability or variance of test statistics for all ACPS tests. By responding with maximum or minimum to the first CEEXEC prompt which is shown in Table E-15, the user can cause CEEXEC to generate a file (on unit 12) containing the maximum or minimum test statistics from corresponding tests in the two test result input files. The user should keep a set of maximum and minimum test result files for each type of ACPS test execution set. Each time a new set of test results is generated, CEEXEC can then be used to update the corresponding maximum and minimum result files. This process should be repeated until all maximum and minimum values do not

change. The maximum and minimum result files can then be compared by CEEXEC to determine the test variance. Only corresponding maximum or minimum result files of different test types should be compared.

E.3 TEST NAME AND FILE NAME ORDER

In order to compare corresponding test records in test statistic input files, each ACPS comparison program assumes that test records (i.e., those records prefixed by the symbol <) appear in a sorted order as follows:

- a. For test and program names, the order is determined by sorting logic applied to character fields in the following sequence:

<u>Character Position</u>	<u>Sorting Order</u>
2	A,F,G,L,O
4	0,1,....,9,A,B,C,D,E,F
5	0,1,....,9,A,B,....,Z
6	0,1,....,9,A,B,....,Z
7	0,1,....,9,A,B,....,Z
3	0,D,I,M,N,S,C

Program names appear on linkage edit test records in compile-time statistic files and test names appear on test records in run-time statistic files.

- b. For file names, the sorting logic is applied to character fields in the following order:

<u>Character Position</u>	<u>Sorting Order</u>
2	A,F,G,L,O
4	P,0,1,....,9,A,B,C,D,E,F
5	0,1,....,9,A,B,....,Z
6	0,1,....,9,A,B,....,Z
7	D,1,....,9,A,B,....,Z
3	J,D,F,M,N

File names appear in compilation and file size test records in compile-time statistic files.

Table E-1. CCOMP Interface File Specifications

FORTRAN		
Input/Output Unit	Type	Description
5	Input	User terminal input
6	Output	Terminal output for prompts and error messages
7	Input	Unformatted compile-time test result input file
8	Input	Unformatted compile-time test result input file
9	Output	Test result comparison output file
10	Output	Formatted output file for input from unit 7
11	Output	Formatted output file for input from unit 8

Table E-2. CEEXEC Interface File Specifications

FORTRAN		
Input/Output Unit	Type	Description
5	Input	User terminal input
6	Output	Terminal output for prompts and error messages
7	Input	Unformatted run-time test result input file
8	Input	Unformatted run-time test result input file
9	Output	Test result comparison output file
10	Output	Formatted output file for input from unit 7
11	Output	Formatted output file for input from unit 8
12	Output	Optionally generated file that contains the minimum or maximum test statistics for corresponding test results input from units 7 and 8

Table E-3. DEC VAX Ada Compile-time Test Statistic File

```

Marking Set /Limit= 150 /Quota= 1500 /Extent= 2500
Adjustment enabled Authorized Quota= 1500 Authorized Extent= 2500
*start samples
*system
Status on 17-DEC-1984 04:24:34.16 Elapsed CPU : 0 00:00:10.52 Buff. I/O : 229 Car. us. : 150 Open files :
Status on 17-DEC-1984 04:24:35.43 Elapsed CPU : 0 00:01:49.03 Buff. I/O : 302 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:24:35.66 Elapsed CPU : 0 00:01:49.92 Buff. I/O : 304 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:24:42.07 Elapsed CPU : 0 00:01:55.51 Buff. I/O : 420 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:24:43.20 Elapsed CPU : 0 00:01:55.69 Buff. I/O : 424 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:24:52.65 Elapsed CPU : 0 00:01:56.43 Buff. I/O : 669 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:24:52.97 Elapsed CPU : 0 00:01:56.61 Buff. I/O : 672 Car. us. : 150 Open files :
Status on 17-DEC-1984 04:27:05.12 Elapsed CPU : 0 00:02:04.74 Buff. I/O : 810 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:27:05.30 Elapsed CPU : 0 00:02:04.94 Buff. I/O : 814 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:27:21.46 Elapsed CPU : 0 00:02:15.06 Buff. I/O : 809 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:27:21.73 Elapsed CPU : 0 00:02:15.24 Buff. I/O : 893 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:27:37.70 Elapsed CPU : 0 00:02:24.09 Buff. I/O : 644 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:27:38.05 Elapsed CPU : 0 00:02:24.29 Buff. I/O : 670 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:27:54.42 Elapsed CPU : 0 00:02:33.41 Buff. I/O : 743 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:27:54.67 Elapsed CPU : 0 00:02:33.41 Buff. I/O : 747 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:28:01.47 Elapsed CPU : 0 00:02:37.89 Buff. I/O : 793 Car. us. : 300 Open files :
*ADDP000
Status on 17-DEC-1984 04:28:01.67 Elapsed CPU : 0 00:02:37.72 Buff. I/O : 797 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:28:11.40 Elapsed CPU : 0 00:02:41.52 Buff. I/O : 834 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:28:11.60 Elapsed CPU : 0 00:02:41.65 Buff. I/O : 836 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:29:14.44 Elapsed CPU : 0 00:03:20.61 Buff. I/O : 1076 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:29:14.62 Elapsed CPU : 0 00:03:20.72 Buff. I/O : 1078 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:29:23.26 Elapsed CPU : 0 00:03:25.14 Buff. I/O : 1120 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:29:23.50 Elapsed CPU : 0 00:03:25.37 Buff. I/O : 1124 Car. us. : 300 Open files :
Status on 17-DEC-1984 04:29:31.04 Elapsed CPU : 0 00:03:29.52 Buff. I/O : 1161 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:29:32.11 Elapsed CPU : 0 00:03:29.60 Buff. I/O : 1165 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:29:52.94 Elapsed CPU : 0 00:03:43.70 Buff. I/O : 1230 Car. us. : 150 Open files :
*ADDP000
Status on 17-DEC-1984 04:29:53.27 Elapsed CPU : 0 00:03:43.94 Buff. I/O : 1243 Car. us. : 300 Open files :
XADAC-M-EMDIACS, Ada compilation completed with 2 diagnostics
Status on 17-DEC-1984 04:30:02.09 Elapsed CPU : 0 00:03:49.41 Buff. I/O : 1289 Car. us. : 150 Open files :
*end samples
*start links
*ADDP000

```

Table E-3. DEC VAX Ada Compile-time Test Statistic File (Continued)

```

Status on 17-DEC-1986 04:30:03.19 Elapsed CPU : 0 00:03:49.54 Buff. I/O : 1293 Cur. ws. : 300 Open files :
ZACS-I-CL LINKING, Invoking the VAX/VMS Linker
in module A400000 file PUBLIC:U15554.ALAM.ADA.DEC1A400000.OBJ:1
Status on 17-DEC-1986 04:30:20.01 Elapsed CPU : 0 00:03:51.26 Buff. I/O : 1346 Cur. ws. : 300 Open files :
<AF05306
Status on 17-DEC-1986 04:30:21.00 Elapsed CPU : 0 00:03:51.58 Buff. I/O : 1350 Cur. ws. : 300 Open files :
ZACS-I-CL LINKING, Invoking the VAX/VMS Linker
in module AF05307 file PUBLIC:U15554.ALAM.ADA.DEC1AF05307.OBJ:1
ZLINK-M-WARNERS, compilation warnings in module AF05306 file PUBLIC:U15554.ALAM.ADA.DEC1AF05306.OBJ:1
ZLINK-M-WARNERS, compilation warnings in module AF05305 file PUBLIC:U15554.ALAM.ADA.DEC1AF05305.OBJ:1
Status on 17-DEC-1986 04:30:39.50 Elapsed CPU : 0 00:03:53.61 Buff. I/O : 1421 Cur. ws. : 150 Open files :
<AD00000
Status on 17-DEC-1986 04:30:39.61 Elapsed CPU : 0 00:03:53.59 Buff. I/O : 1425 Cur. ws. : 150 Open files :
ZACS-I-CL LINKING, Invoking the VAX/VMS Linker
Status on 17-DEC-1986 04:30:50.53 Elapsed CPU : 0 00:03:55.62 Buff. I/O : 1404 Cur. ws. : 150 Open files :
<AD00000
Status on 17-DEC-1986 04:30:50.79 Elapsed CPU : 0 00:03:55.77 Buff. I/O : 1406 Cur. ws. : 150 Open files :
ZACS-I-CL LINKING, Invoking the VAX/VMS Linker
Status on 17-DEC-1986 04:31:17.10 Elapsed CPU : 0 00:03:57.71 Buff. I/O : 1545 Cur. ws. : 150 Open files :
<AD00506
Status on 17-DEC-1986 04:31:17.42 Elapsed CPU : 0 00:03:57.88 Buff. I/O : 1553 Cur. ws. : 300 Open files :
ZACS-I-CL LINKING, Invoking the VAX/VMS Linker
in module A000505 file PUBLIC:U15554.ALAM.ADA.DEC1A000505.OBJ:1
ZLINK-M-WARNERS, compilation warnings in module A000504 file PUBLIC:U15554.ALAM.ADA.DEC1A000504.OBJ:1
ZLINK-M-WARNERS, compilation warnings
Status on 17-DEC-1986 04:31:34.91 Elapsed CPU : 0 00:03:59.69 Buff. I/O : 1612 Cur. ws. : 150 Open files :
cond links
total source size
system 109
<A400000 2
<A400000 4
<AF00000 20
<AF05305 6
<AF05306 7
<AF05307 7
<AF05308 3
<AC00000 2
<AC00000 17
<AD00000 2
<AD00000 4
<AD00505 8
<AD00506 4
cond source size = 215
total object size
system 368
<A400000 7
<A400000 10
<AF00000 62
<AF05305 17
<AF05306 20
<AF05307 20
<AF05308 8
<AC00000 7
<AC00000 122
<AD00000 8
<AD00000 13

```

Table E-3. DEC VAX Ada Compile-time Test Statistic File (Concluded)

```
<A000505 26
<A000506 11
<end object size = 699
<start executable size
<AA00000 59
<AF05308 64
<AG00008 68
<AGM0008 67
<A000506 64
<end executable size = 322
```

Table E-4. ECSPO JOVIAL Compile-time Test Statistic File

```

Working Set /Limit: 150 /Queue: 1500 /Extent: 2500
Argument enabled Authorized Queue: 1500 Authorized Extent: 2500
start compile
system
Status on 17-DEC-1966 04:16:26.60 Elapsed CPU : 0 00:00:06.47 Buff. I/O : 146 Cur. no. : 150 Open files :
File Usage Creation Date/Time
-----
HSC0000DUAS: (JOVIAL.COMTIC.TOOLS.FV.EXT JOVIAL.EXT) Compiler Image 15-FEB-1966 23:20:13
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CPL:04 Source Module 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.OBJ:11 Object Output 17-DEC-1966 04:16:36
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CMP:25 Compile Output 17-DEC-1966 04:16:37
256 Source Lines. 145 JOVIAL Statements.
NO MESSAGES.
Compilation Time: CPU TIME = 0:00:07.31 ELAPSED TIME = 00:00:11.20
File Usage Creation Date/Time
-----
HSC0000DUAS: (JOVIAL.COMTIC.TOOLS.FV.EXT JOVIAL.EXT) Compiler Image 15-FEB-1966 23:20:13
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.JOV:121 Source Module 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CMP:23 Compile Input 17-DEC-1966 04:16:37
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.CMP:13 Compile Input 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.OBJ:11 Object Output 17-DEC-1966 04:17:21
984 Source Lines. 619 JOVIAL Statements.
3 MESSAGES: 3 WARNING
Compilation Time: CPU TIME = 0:00:46.07 ELAPSED TIME = 00:00:45.06
File Usage Creation Date/Time
-----
HSC0000DUAS: (JOVIAL.COMTIC.TOOLS.FV.EXT JOVIAL.EXT) Compiler Image 15-FEB-1966 23:20:13
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.CPL:16 Source Module 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CMP:23 Compile Input 17-DEC-1966 04:16:37
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.OBJ:11 Object Output 17-DEC-1966 04:17:33
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.CMP:14 Compile Output 17-DEC-1966 04:17:33
60 Source Lines. 42 JOVIAL Statements.
NO MESSAGES.
Compilation Time: CPU TIME = 0:00:04.55 ELAPSED TIME = 00:00:08.15
File Usage Creation Date/Time
-----
HSC0000DUAS: (JOVIAL.COMTIC.TOOLS.FV.EXT JOVIAL.EXT) Compiler Image 15-FEB-1966 23:20:13
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.JOV:19 Source Module 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CMP:23 Compile Input 17-DEC-1966 04:16:37
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.CMP:14 Compile Input 17-DEC-1966 04:17:33
PUBLIC: (UISS54.ALAN.JOVIAL.JOVSYS.OBJ:11 Object Output 17-DEC-1966 04:17:47
50 Source Lines. 21 JOVIAL Statements.
NO MESSAGES.
Compilation Time: CPU TIME = 0:00:06.02 ELAPSED TIME = 00:00:12.70
Status on 17-DEC-1966 04:17:54.34 Elapsed CPU : 0 00:01:15.45 Buff. I/O : 445 Cur. no. : 150 Open files :
cJADP000
Status on 17-DEC-1966 04:17:54.57 Elapsed CPU : 0 00:01:15.06 Buff. I/O : 447 Cur. no. : 150 Open files :
File Usage Creation Date/Time
-----
HSC0000DUAS: (JOVIAL.COMTIC.TOOLS.FV.EXT JOVIAL.EXT) Compiler Image 15-FEB-1966 23:20:13
PUBLIC: (UISS54.ALAN.JOVIAL.JADP000.CPL:17 Source Module 16-DEC-1966 16:01:04
PUBLIC: (UISS54.ALAN.JOVIAL.JOURJOV.CMP:23 Compile Input 17-DEC-1966 04:16:37
PUBLIC: (UISS54.ALAN.JOVIAL.JADP000.OBJ:11 Object Output 17-DEC-1966 04:18:03
PUBLIC: (UISS54.ALAN.JOVIAL.JADP000.CMP:13 Compile Output 17-DEC-1966 04:18:04

```

Table E-4. ECSP0 JOVIAL Compile-time Test Statistic File (Continued)

```

WSC00010UAS: (JOVIAL.CONFIG.TOOLS.FV.(XHE JOVIAL.(XHE:)) Compiler Image      15-FEB-1966 23:20:13
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CPL:14 Source Module      15-DEC-1966 04:16:53
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:23 Compiler Input      17-DEC-1966 04:16:57
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.OBJ:11 Object Output      17-DEC-1966 04:21:40
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:18 Compiler Input      17-DEC-1966 04:21:40
34 Source Lines. 20 JOVIAL Statements.
NO MESSAGES.
Compilation Time: CPU TIME = 0:00:05.15 ELAPSED TIME = 00:00:06.06
Status on 17-DEC-1966 04:21:42.90 Elapsed CPU : 0 00:03:34.50 Buff. I/O : 1944 Cur. no. : 150 Open files :
CJ000505
Status on 17-DEC-1966 04:21:43.27 Elapsed CPU : 0 00:03:34.71 Buff. I/O : 1948 Cur. no. : 300 Open files :
File Usage Creation Date/Time
-----
WSC00010UAS: (JOVIAL.CONFIG.TOOLS.FV.(XHE JOVIAL.(XHE:)) Compiler Image      15-FEB-1966 23:20:13
PUBLIC: (UI5554.ALAN.JOVIAL.IJ000505.JOV:23 Source Module      15-DEC-1966 17:11:55
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:23 Compiler Input      17-DEC-1966 04:16:57
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:18 Compiler Input      17-DEC-1966 04:21:40
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:14 Compiler Input      17-DEC-1966 04:17:53
PUBLIC: (UI5554.ALAN.JOVIAL.IJ000505.OBJ:11 Object Output      17-DEC-1966 04:21:54
92 Source Lines. 47 JOVIAL Statements.
9 MESSAGES: 9 INFORMATION
Compilation Time: CPU TIME = 0:00:10.00 ELAPSED TIME = 00:00:10.09
Status on 17-DEC-1966 04:21:50.99 Elapsed CPU : 0 00:03:47.79 Buff. I/O : 2025 Cur. no. : 150 Open files :
CJ000506
Status on 17-DEC-1966 04:21:59.23 Elapsed CPU : 0 00:03:47.95 Buff. I/O : 2029 Cur. no. : 300 Open files :
File Usage Creation Date/Time
-----
WSC00010UAS: (JOVIAL.CONFIG.TOOLS.FV.(XHE JOVIAL.(XHE:)) Compiler Image      15-FEB-1966 23:20:13
PUBLIC: (UI5554.ALAN.JOVIAL.IJ000506.JOV:14 Source Module      15-DEC-1966 17:11:59
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:23 Compiler Input      17-DEC-1966 04:16:57
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:18 Compiler Input      17-DEC-1966 04:21:40
PUBLIC: (UI5554.ALAN.JOVIAL.IJ00P000.CMP:14 Compiler Input      17-DEC-1966 04:17:53
PUBLIC: (UI5554.ALAN.JOVIAL.IJ000506.OBJ:11 Object Output      17-DEC-1966 04:22:00
78 Source Lines. 40 JOVIAL Statements.
9 MESSAGES: 9 INFORMATION
Compilation Time: CPU TIME = 0:00:06.39 ELAPSED TIME = 00:00:10.34
Status on 17-DEC-1966 04:22:11.20 Elapsed CPU : 0 00:03:55.53 Buff. I/O : 2104 Cur. no. : 150 Open files :
card compiler
card linker
CJ000000
Status on 17-DEC-1966 04:22:11.50 Elapsed CPU : 0 00:03:55.47 Buff. I/O : 2110 Cur. no. : 300 Open files :
Status on 17-DEC-1966 04:22:20.05 Elapsed CPU : 0 00:03:59.07 Buff. I/O : 2109 Cur. no. : 150 Open files :
CJ000500
Status on 17-DEC-1966 04:22:21.12 Elapsed CPU : 0 00:04:00.00 Buff. I/O : 2193 Cur. no. : 300 Open files :
Status on 17-DEC-1966 04:22:30.04 Elapsed CPU : 0 00:04:05.01 Buff. I/O : 2200 Cur. no. : 150 Open files :
CJ000000
Status on 17-DEC-1966 04:22:30.79 Elapsed CPU : 0 00:04:05.15 Buff. I/O : 2292 Cur. no. : 300 Open files :
Status on 17-DEC-1966 04:22:40.03 Elapsed CPU : 0 00:04:10.12 Buff. I/O : 2302 Cur. no. : 150 Open files :
CJ000000
Status on 17-DEC-1966 04:22:40.37 Elapsed CPU : 0 00:04:10.27 Buff. I/O : 2307 Cur. no. : 300 Open files :
Status on 17-DEC-1966 04:22:49.61 Elapsed CPU : 0 00:04:15.31 Buff. I/O : 2477 Cur. no. : 150 Open files :
CJ000506
Status on 17-DEC-1966 04:22:49.91 Elapsed CPU : 0 00:04:15.20 Buff. I/O : 2481 Cur. no. : 300 Open files :
Status on 17-DEC-1966 04:22:50.53 Elapsed CPU : 0 00:04:19.72 Buff. I/O : 2555 Cur. no. : 150 Open files :
card linker

```

Table E-4. ECSP0 JOVIAL Compile-time Test Statistic File (Concluded)

```
<start source size
<system 104
<JA0P000 2
<JA00000 3
<JF0P000 21
<JF05305 7
<JF05306 9
<JF05307 9
<JF05308 3
<JG0P000 2
<JG00008 43
<JGM0008 2
<J00P000 4
<J000505 8
<J000506 4
<end source size = 221
<start object size
<system 52
<JA0P000 3
<JA00000 2
<JF0P000 23
<JF05305 5
<JF05306 6
<JF05307 6
<JF05308 1
<JG0P000 3
<JG00008 20
<JGM0008 1
<J00P000 4
<J000505 5
<J000506 2
<end object size = 133
<start executable size
<JA00000 29
<JF05308 31
<JG00008 32
<JGM0008 32
<J000506 31
<end executable size = 155
```


Table E-5. ACPS Compilation Comparator Formatted Ada File

```

      COMPILATION - SHOW STATUS STATISTICS(UNIT 8)

FILES(S) WALL-CLOCK CPU-TIME  BUFIO  DIRIO PAGE-FAULTS

system      121.27    99.31    153    238    13661
AADP000      7.31     3.59     36     49    1982
AA00000      9.37     4.74     45     60    2281
AF0P000     12.15     8.11     37     58    2521
AF05305     16.08     8.11     75    113    3175
AF05306     16.05     8.85     73    100    3215
AF05307     16.37     8.95     73    104    3305
AF05308      8.80     4.18     46     60    2168
AG0P000      7.73     3.80     37     53    2574
AG00008     62.84    38.96    240    335   15671
AGM0008      8.64     4.42     42     59    2277
ADDP000      8.26     4.15     37     63    2440
AD00505     20.85    14.10     73    109    3636
AD00506      9.62     5.47     46     62    2306

totals      325.34   216.74   1013   1463   61212
  
```

```

      LINKAGE EDIT - SHOW STATUS STATISTICS(UNIT 8)

FILES(S) WALL-CLOCK CPU-TIME  BUFIO  DIRIO PAGE-FAULTS

AA00000     17.62     1.70     53     49    489
AF05308     18.50     2.03     71     58    484
AG00008     18.72     2.03     59     49    701
AGM0008     18.31     1.94     60     51    493
AD00506     17.49     1.81     59     49    499

totals      90.64     9.51    302    256    2666
  
```

Table E-5. ACPS Compilation Comparator Formatted Ada File (Continued)

SOURCE MODULE- FILE SIZE STATISTICS(UNIT 8)

FILE(S) SIZE(BLOCKS)

system	109
AA0P000	2
AA00000	4
AF0P000	20
AF05305	6
AF05306	7
AF05307	7
AF05308	3
AG0P000	2
AG00008	37
AGM0008	2
A00P000	4
A000505	8
A000506	4
totals	215

OBJECT MODULE- FILE SIZE STATISTICS(UNIT 8)

FILE(S) SIZE(BLOCKS)

system	368
AA0P000	7
AA00000	10
AF0P000	62
AF05305	17
AF05306	20
AF05307	20
AF05308	8
AG0P000	7
AG00008	122
AGM0008	8
A00P000	13
A000505	26
A000506	11
totals	699

Table E-5. ACPS Compilation Comparator Formatted Ada File (Concluded)

LOAD MODULE - FILE SIZE STATISTICS(UNIT 8)	
FILE(S)	SIZE(BLOCKS)
AA00000	59
AF05308	64
AG00008	68
AGM0008	67
AD00506	64
totals	322

Table E-6. ACPS Compilation Comparator Formatted JOVIAL File

```

COMPILATION - SHOW STATUS STATISTICS(UNIT 7)

FILES(S) WALL-CLOCK CPU-TIME  BUFIO  DIRIO PAGE-FAULTS

system      89.76    68.98    319    302    6410
JA0P000      9.65     5.07     73     56    1064
JA00000     10.47     6.05     78     57    1323
JF0P000     21.81    17.00     73     62    1093
JF05305     21.28    12.68    149    105    2472
JF05306     22.69    13.99    150    109    3071
JF05307     22.87    13.80    149    106    2537
JF05308      9.80     5.67     82     51    1386
JG0P000      9.35     5.17     73     54    1120
JG00008     75.97    48.44    470    328    8218
JGM0008     10.23     5.78     82     50    1326
JO0P000     10.43     6.10     73     52    1293
JO00505     15.72    11.08     77     55    1436
JO00506     11.97     7.40     77     51    1212

totals      342.00    227.21    1925    1438    33961

```

```

LINKAGE EDIT - SHOW STATUS STATISTICS(UNIT 7)

FILES(S) WALL-CLOCK CPU-TIME  BUFIO  DIRIO PAGE-FAULTS

JA00000      9.35     4.40     79    140     696
JF05308      9.42     5.01     95    137     742
JG00008      9.24     4.97     90    137     747
JGM0008      9.24     4.84     90    137     722
JO00506      8.62     4.44     74    127     828

totals      45.87    23.66    428    678    3735

```

Table E-6. ACPS Compilation Comparator Formatted JOVIAL File (Continued)

SOURCE MODULE- FILE SIZE STATISTICS(UNIT 7)

FILE(S) SIZE(BLOCKS)

system	104
JAOP000	2
JA00000	3
JFOP000	21
JF05305	7
JF05306	9
JF05307	9
JF05308	3
JG0P000	2
JG00008	43
JGM0008	2
JOOP000	4
JO00505	8
JO00506	4
totals	221

OBJECT MODULE- FILE SIZE STATISTICS(UNIT 7)

FILE(S) SIZE(BLOCKS)

system	52
JAOP000	3
JA00000	2
JFOP000	23
JF05305	5
JF05306	6
JF05307	6
JF05308	1
JG0P000	3
JG00008	20
JGM0008	1
JOOP000	4
JO00505	5
JO00506	2
totals	133

Table E-6. ACPS Compilation Comparator Formatted JOVIAL File (Concluded)

LOAD MODULE - FILE SIZE STATISTICS(UNIT 7)	
FILE(S)	SIZE(BLOCKS)
JA00000	29
JF05308	31
JG00008	32
JGM0008	32
JO00506	31
totals	155

Table E-7. ACPS Compilation Comparator Comparison Output File

The Aerospace Corporation

input test identification(< 55 characters):

>

VAX Ada vs ECSPD JOVIAL(opt on)

which operating system(vms,...)?

>

vms

Table E-7. ACPS Compilation Comparator Comparison Output File (Continued)

VAX Ada vs ECSPD JOVIAL(apt on)

FILES(S)	WALL-CLOCK	CPU-TIME	BUFIO	DIRIO	PAGE-FAULTS
system	1.35	1.44	0.48	0.79	2.13
AAOP000	0.76	0.71	0.49	0.88	1.86
AAO0000	0.89	0.78	0.58	1.05	1.72
AFOP000	0.56	0.68	0.51	0.94	2.31
AF05305	0.76	0.64	0.50	1.08	1.28
AF05306	0.71	0.63	0.49	0.92	1.05
AF05307	0.72	0.65	0.49	0.98	1.30
AF05308	0.90	0.74	0.56	1.18	1.56
AGOP000	0.83	0.74	0.51	0.98	2.30
AG00008	0.83	0.80	0.51	1.02	1.91
AGH0008	0.84	0.76	0.51	1.18	1.72
ADOP000	0.79	0.68	0.51	1.21	1.89
AD00505	1.33	1.27	0.95	1.98	2.53
AD00506	0.80	0.74	0.60	1.22	1.90

COMPILATION totals:

0.95	0.95	0.53	1.02	1.80
COMPILATION	COMPARISONS	UNIT 8 / UNIT 71		

Table E-7. ACPS Compilation Comparator Comparison Output File (Continued)

VAX Ada vs ECSPD JOVIAL (opt on)

FILES(S) WALL-CLOCK CPU-TIME BUFIO DIRIO PAGE-FAULTS

AA00000	1.88	0.39	0.67	0.35	0.70
AF05308	1.96	0.41	0.75	0.42	0.65
AG00008	2.03	0.41	0.66	0.36	0.94
AG00008	1.98	0.40	0.67	0.37	0.68
AD00506	2.03	0.41	0.80	0.39	0.60

LINKAGE EDIT totals:

1.98	0.40	0.71	0.38	0.71
------	------	------	------	------

LINKAGE EDIT COMPARISONS (UNIT 8 / UNIT 7)

Table E-7. ACPS Compilation Comparator Comparison Output File (Continued)

VAX Ada vs ECSPD JOVIAL(apt on)

FILE(S)	SIZE(BLOCKS)
system	1.05
AA0P000	1.00
AA00000	1.33
AF0P000	0.95
AF05305	0.04
AF05306	0.78
AF05307	0.78
AF05308	1.00
AG0P000	1.00
AG00000	0.04
AGM0000	1.00
AD0P000	1.00
AD00505	1.00
AD00506	1.00

SOURCE MODULE totals:

0.97

SOURCE MODULE FILE SIZE COMPARISONS: UNIT 6 / UNIT 71

Table E-7. ACPS Compilation Comparator Comparison Output File (Continued)

VAX Ada vs ECSPD JOVIAL(eps) m)

FILE(S)	SIZE(BLOCKS)
system	7.00
AA0P000	2.33
AA00000	5.00
AF0P000	2.70
AF05105	3.40
AF05106	3.33
AF05107	3.33
AF05108	0.00
AG0P000	2.33
AG00000	6.10
AG00008	0.00
AD0P000	3.25
AD00505	5.20
AD00506	5.50

OBJECT MODULE totals:

5.26

Table E-7. ACPS Compilation Comparator Comparison Output File (Concluded)

```

VAX Ada vs ECSPD JOVIAL(opt on)

FILE(S)      SIZE(BLOCKS)

AA00000      2.03
AF05100      2.06
AC00000      2.13
AD00000      2.09
AD00500      2.06

LOAD MODULE  totals:

                2.00

```

Table E-8. DEC VAX Ada Run-time Test Statistic File

```

Marking Set /Limit= 850 /Quota= 850 /Extent= 2500
Adjustment disabled Authorized Quota= 3500 Authorized Extent= 2500
Status on 24-DEC-1986 04:53:21.19 Elapsed CPU : 0 00:17:37.75 Buff. I/O : 2922 Cur. no. : 850 Open files :
<A000000 0.000000 1 0 0 0 0 3 317640 007623 277 90
1 -1
<A000000 3.319946 9990 332 0 0 0 3 317640 007623 277 90
<A000000 0.000213 0 30 0 0 0 3 317640 007623 277 90
2 216
Status on 24-DEC-1986 04:53:28.07 Elapsed CPU : 0 00:17:43.52 Buff. I/O : 2981 Cur. no. : 850 Open files :
<A705305 0.000000 1 0 0 0 0 3 310976 007623 277 99
1 -1
<A705305 3.559940 7690 354 0 0 0 3 310976 007623 277 99
<A705306 0.000000 1 0 0 0 0 3 310976 007623 277 99
1 -1
<A705306 5.739940 7690 573 0 0 0 3 310976 007623 277 99
<A755306 2.379943 7690 217 0 0 0 0 0 0 0 0
1 -1
<A705307 0.000000 1 0 0 0 0 3 310976 007623 277 99
1 -1
<A705307 5.019946 7690 202 0 0 0 3 310976 007623 277 99
<A755307 2.259949 7690 224 0 0 0 0 0 0 0 0
1 -1
<A000000 0.000000 1 0 0 0 0 3 316600 007623 223 65
<A000000 1.419903 1290 342 225 0 0 3 327540 007623 243 50
<A000000 0.001211 0 30 15 0 0 3 316576 007623 255 102
2 1231
<A000507 0.000000 1 0 0 0 0 3 316600 007623 297 00
1 -1
<A000507 0.439941 9990 45 0 0 0 3 316600 007623 297 00

*** Elapsed time in above test is too short (< 100 times clock granularity)
<A000500 0.000000 1 0 0 0 0 3 316600 007623 297 00
1 -1
<A000500 3.759949 9990 375 0 0 0 3 316600 007623 297 00
<A0C0500 0.566676 0 033 100 100 100 100 100 100 100
1 -1
Status on 24-DEC-1986 04:54:01.22 Elapsed CPU : 0 00:18:10.39 Buff. I/O : 3144 Cur. no. : 850 Open files :

```

Table E-9. ECSP0 JOVIAL Run-time Test Statistic File

```

Working Set /Limit: 850 /Quota: 850 /Extent: 2500
Adjustment disabled Authorized Quota: 1500 Authorized Extent: 2500
Status on 74-DEC-1964 04:54:06.27 Elapsed CPU : 0 00:10:12.03 Buff. I/O : 3216 Cur. no. : 850 Open files :
CJ000000 0.000000 1 0 0 0 0 3 119296 005007 259 67
CJ000000 1.030000 9998 193 0 0 0 3 119296 005007 259 67
CJ000000 0.000203 0 20 0 0 0 3 119296 005007 259 67
JOVIAL STOP -- TERMINATION CODE IS 0
Status on 74-DEC-1964 04:54:12.28 Elapsed CPU : 0 00:10:16.95 Buff. I/O : 3275 Cur. no. : 850 Open files :
CJ005305 0.000000 1 0 0 0 0 3 120320 005007 270 69
CJ005305 2.570000 7998 257 0 0 0 3 120320 005007 270 69
CJ005306 0.000000 1 0 0 0 0 3 120320 005007 270 69
CJ005306 3.020000 7998 302 0 0 0 3 120320 005007 270 69
CJ005306 0.450000 7998 45 0 0 0 0 0 0 0 0
*** elapsed time in above test is too short( 100 times clock granularity)
CJ005307 0.000000 1 0 0 0 0 3 120320 005007 270 69
CJ005307 2.040000 7998 204 0 0 0 3 120320 005007 270 69
CJ005307 0.209999 7998 29 0 0 0 0 0 0 0 0
*** elapsed time in above test is too short( 100 times clock granularity)
JOVIAL STOP -- TERMINATION CODE IS 0
CJ000000 0.000000 1 0 4 0 0 3 131072 005007 272 58
CJ000000 0.050000 1798 85 0 0 0 3 131072 005007 272 58
*** elapsed time in above test is too short( 100 times clock granularity)
JOVIAL STOP -- TERMINATION CODE IS 0
CJ000000 0.000000 0 10 4 0 0 3 131072 005007 280 57
CJ000000 2.336000 841 0 0 0 3 120320 005007 269 61
JOVIAL STOP -- TERMINATION CODE IS 0
CJ000507 0.000000 1 0 0 0 0 3 120320 005007 269 61
CJ000507 0.690000 9998 69 0 0 0 3 120320 005007 269 61
*** elapsed time in above test is too short( 100 times clock granularity)
CJ000508 0.000000 1 0 0 0 0 3 120320 005007 269 61
CJ000508 2.780000 9998 278 0 0 0 3 120320 005007 269 61
CJ000508 4.020906 0 402 100 100 100 100 100 100 100
JOVIAL STOP -- TERMINATION CODE IS 0
Status on 74-DEC-1964 04:54:36.09 Elapsed CPU : 0 00:10:25.14 Buff. I/O : 3420 Cur. no. : 850 Open files :

```

Table E-10. CEEXEC Formatted VAX Ada Input File without VMS Statistics

VAX Ada vs ECSPD JOVIAL(opt on,no VMS)			BENCHMARK TEST RESULTS: ADA					UNIT(8)
TEST-ID	WALL-TIME	ITERATIONS	RUN STATISTICS(0*=NOT APPLICABLE)					
AA00000	0.000000	1	0*	0*	0*	0*	0*	0*
AA00000	3.319946	99998	0*	0*	0*	0*	0*	0*
AA00000	0.000213	0	263	214	0*	0*	0*	0*
AF05305	0.000000	1	0*	0*	0*	0*	0*	0*
AF05305	1.070055	74998	0*	0*	0*	0*	0*	0*
AF05306	0.000000	1	0*	0*	0*	0*	0*	0*
AF05306	3.250047	74998	0*	0*	0*	0*	0*	0*
AFS5306	2.179993	74998	0*	0*	0*	0*	0*	0*
AF05307	0.000000	1	0*	0*	0*	0*	0*	0*
AF05307	3.330003	74998	0*	0*	0*	0*	0*	0*
AFS5307	2.259949	74998	0*	0*	0*	0*	0*	0*
AG00008	0.000000	1	0*	0*	0*	0*	0*	0*
AG00008	1.376889	1298	0*	0*	0*	0*	0*	0*
AGM0008	0.000998	0	3264	1017	0*	0*	0*	0*
AO00507	0.000000	1	0*	0*	0*	0*	0*	0*
AO00507	0.108006	9998	0*	0*	0*	0*	0*	0*
**** elapsed time in above test is too short								
AO00508	0.000000	1	0*	0*	0*	0*	0*	0*
AO00508	3.428014	9998	0*	0*	0*	0*	0*	0*
AOC0508	8.546476	0	0*	0*	0*	0*	0*	0*

Table E-11. CEEXEC Formatted ECSPO JOVIAL Input File without VMS Statistics

VAX Ada vs ECSPO JOVIAL(opt on,no VMS)

BENCHMARK TEST RESULTS: JOVIAL UNIT(7)

TEST-ID	WALL-TIME	ITERATIONS	RUN STATISTICS(0*=NOT APPLICABLE)					
JA00000	0.000000	1	22	0*	0*	0*	0*	
JA00000	1.930000	99998	0*	0*	0*	0*	0*	
JAM0000	0.000203	0	218	214	0*	0*	0*	
JF05305	0.000000	1	137	0*	0*	0*	0*	
JF05305	1.122510	74998	0*	0*	0*	0*	0*	
JF05306	0.000000	1	228	0*	0*	0*	0*	
JF05306	1.572510	74998	0*	0*	0*	0*	0*	
JFS5306	0.450000	74998	91	0*	0*	0*	0*	
**** elapsed time in above test is too short								
JF05307	0.000000	1	227	0*	0*	0*	0*	
JF05307	1.412510	74998	0*	0*	0*	0*	0*	
JFS5307	0.289999	74998	90	0*	0*	0*	0*	
**** elapsed time in above test is too short								
JG00008	0.000000	1	0*	0*	0*	0*	0*	
JG00008	0.824948	1298	0*	0*	0*	0*	0*	
**** elapsed time in above test is too short								
JGM0008	0.000637	0	2178	627	0*	0*	0*	
JO00507	0.000000	1	59	0*	0*	0*	0*	
JO00507	0.497035	9998	0*	0*	0*	0*	0*	
**** elapsed time in above test is too short								
JO00508	0.000000	1	683	0*	0*	0*	0*	
JO00508	2.587035	9998	0*	0*	0*	0*	0*	
JOC0508	4.028986	0	870	0*	0*	0*	0*	

Table E-12. CEXEC Comparison Output File without VMS Statistics

The Aerospace Corporation

```
read operating system dependent status records(yes/no)?  
>  
no  
read operating system dependent fields(yes/no)?  
>  
no  
input test title  
>  
VAX Ada vs ECSPD JOVIAL(opt on,no VMS)  
input minimum test time in seconds(e.g. 1.1):  
>  
1.0
```

Table E-12. CEXEC Comparison Output File without VMS Statistics
(Continued)

VAX Ada vs ECSP0 JOVIAL (opt on, no VMS)						
BENCHMARK TEST COMPARISON: UNIT 01ADA 1 / UNIT 71JOVIAL 1						
TEST-ID	WALL-TIME	RUN STATISTICS (1.0=NOT APPLICABLE)				
AA00000	1.00	1.00	1.00	1.00	1.00	1.00
AA00000	1.72	1.00	1.00	1.00	1.00	1.00
AA00000	1.05	1.21	1.00	1.00	1.00	1.00
AF05305	1.00	1.00	1.00	1.00	1.00	1.00
AF05305	0.95	1.00	1.00	1.00	1.00	1.00
AF05306	1.00	1.00	1.00	1.00	1.00	1.00
AF05306	2.07	1.00	1.00	1.00	1.00	1.00
AF05306	4.04	1.00	1.00	1.00	1.00	1.00
AF05307	1.00	1.00	1.00	1.00	1.00	1.00
AF05307	2.36	1.00	1.00	1.00	1.00	1.00
AF05307	7.79	1.00	1.00	1.00	1.00	1.00
AG00000	1.00	1.00	1.00	1.00	1.00	1.00
AG00000	1.67	1.00	1.00	1.00	1.00	1.00
AG00000	1.57	1.50	1.62	1.00	1.00	1.00
AO00507	1.00	1.00	1.00	1.00	1.00	1.00
AO00507	0.22	1.00	1.00	1.00	1.00	1.00
AO00508	1.00	1.00	1.00	1.00	1.00	1.00
AO00508	1.33	1.00	1.00	1.00	1.00	1.00
AO00508	2.12	1.00	1.00	1.00	1.00	1.00
test comparison totals:						
micro	1.57	1.50	1.62	1.00	1.00	1.00
1 iter	1.00	1.00	1.00	1.00	1.00	1.00
>1 iter	1.52	1.00	1.00	1.00	1.00	1.00
diff.	0.00	1.00	1.00	1.00	1.00	1.00

Table E-13. EXEC Formatted VAX Ada Input File with VMS Statistics

VAX Ada vs ECSPD JOVIAL (impl on)			BENCHMARK TEST RESULTS: ADA (UNIT 8)									
TEST-ID	WALL-TIME	ITERATIONS	RUN STATISTICS (00=NOT APPLICABLE)					VMS STATISTICS				
			CPUTIM	PAGEFLY	BUVIO	DIRIO		FILENT	POSPACE	PISPACE	PPGENT	SPGENT
AA00000	0.000000	1	00	00	00	00	00	3	317440	007423	277	90
AA00000	3.319446	99998	00	00	00	00	00	3	317440	007423	277	90
AA00000	0.000213	0	263	332	00	00	00	3	317440	007423	277	90
AF05305	0.000000	1	00	214	00	00	00	3	317440	007423	277	90
AF05305	0.000000	1	00	50	00	00	00	0	1536	0	0	1
AF05305	1.070055	74998	00	00	00	00	00	0	1536	0	0	1
AF05306	0.000000	1	00	108	00	00	00	0	1536	0	0	1
AF05306	0.000000	1	00	00	00	00	00	0	1536	0	0	1
AF05306	3.250047	74998	00	00	00	00	00	0	1536	0	0	1
AF05306	2.179993	74998	00	325	00	00	00	0	1536	0	0	1
AF05307	0.000000	1	00	217	00	00	00	0	0	0	0	0
AF05307	0.000000	1	00	00	00	00	00	0	1536	0	0	1
AF05307	3.330003	74998	00	00	00	00	00	0	1536	0	0	1
AF05307	2.259949	74998	00	334	00	00	00	0	1536	0	0	1
AG00000	0.000000	1	00	226	00	00	00	0	0	0	0	0
AG00000	1.376889	1298	00	00	00	00	00	0	70048	0	44	-33
AG00000	0.000998	0	3264	110	00	00	00	0	209920	0	264	-40
AG00000	0.000998	0	3264	1017	00	00	00	0	70336	0	70	4
AG00507	0.000000	1	00	00	00	00	00	0	2048	0	20	-10
AG00507	0.108006	9998	00	00	00	00	00	0	2048	0	20	-10
**** elapsed time in above test is too short												
AG00508	0.000000	1	00	00	00	00	00	0	2048	0	20	-10
AG00508	1.428014	9998	00	00	00	00	00	0	2048	0	20	-10
AG00508	0.546476	0	00	342	00	00	00	0	2048	0	20	-10
			033	100	100	100	100	100	100	100	100	100

Table E-14. CEEXEC Formatted ECSPO JOVIAL Output File with VMS Statistics

VAX Ada vs ECSPO JOVIAL (opt on)			BENCHMARK TEST RESULTS: JOVIAL (UNIT 7)									
TEST-ID	WALL-TIME	ITERATIONS	RUN STATISTICS (0=NOT APPLICABLE)				VMS STATISTICS					
			CPUTIM	PAGEFLT	BUFIO	DIRIO	FILCNT	PUSPACE	PISPACE	PPGCNT	SPGCNT	
J400000	0.000000	1	22	0	0	0	0	0	0	0	0	
J400000	1.930000	9998	0	0	0	0	3	119296	005087	259	67	
J400000	0.000203	0	218	193	0	0	3	119296	005087	259	67	
J705305	0.000000	1	137	214	0	0	3	119296	005087	259	67	
J705305	1.122510	74998	0	20	0	0	0	1024	0	11	2	
J705306	0.000000	1	228	113	0	0	0	1024	0	11	2	
J705306	1.572510	74998	0	0	0	0	0	1024	0	11	2	
J735306	0.450000	74998	91	158	0	0	0	1024	0	11	2	
*** elapsed time in above test is too short												
J705307	0.000000	1	227	45	0	0	0	0	0	0	0	
J705307	1.412510	74998	0	0	0	0	0	1024	0	11	2	
J735307	0.289999	74998	90	142	0	0	0	1024	0	11	2	
*** elapsed time in above test is too short												
J800000	0.000000	1	0	0	0	0	0	0	0	0	0	
J800000	0.024948	1298	0	4	0	0	0	11776	0	13	-9	
*** elapsed time in above test is too short												
J000008	0.000637	0	2178	627	0	0	0	11776	0	-1	-10	
J000507	0.000000	1	59	-10	0	0	0	1024	0	10	-4	
J000507	0.497035	9998	0	0	0	0	0	1024	0	10	-4	
*** elapsed time in above test is too short												
J000508	0.000000	1	683	0	0	0	0	1024	0	10	-4	
J000508	2.587035	9998	0	0	0	0	0	1024	0	10	-4	
J000508	4.028984	0	870	259	0	0	0	1024	0	10	-4	
			402	100	100	100	100	100	100	100	100	

Table E-15. CEEXEC Comparison Output File with VMS Statistics

```
read operating system dependant status records(yes/no)?
>
yes
read operating system dependant fields(yes/no)?
>
yes
subtract test routine overhead from operating system stats(yes/no)?
>
yes
which operating system(vms,...)?
>
vms
input test title
>
VAX Ada vs ECSPD JOVIAL(opt on)
input minimum test time in seconds(e.g. 1.1): .
>
1.0
```

Table E-15. CEEXEC Comparison Output File with VMS Statistics (Continued)

VAX Ada vs ECSP0 JOVIAL(opt on)							BENCHMARK TEST COMPARISONS: UNIT SIADA 1 / UNIT 71JOVIAL 1								
TEST-ID	WALL-TIME	RUN STATISTICS(1.00=NOT APPLICABLE)						VMS STATISTICS							
		CPUTIM	PAGEFLY	BUFIO	DIRIO	FILENT	POSPACE	PISPACE	PPGENT	SPGENT					
AA00000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.66	1.00	1.07	1.46
AA00000	1.72	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.66	1.00	1.07	1.46
AA00000	1.05	1.21	1.00	1.00	1.00	1.00	2.50	1.00	1.00	1.00	1.00	2.66	1.00	1.07	1.46
AF05105	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF05105	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF05106	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF05106	2.07	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF35106	4.04	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AF05107	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF05107	2.36	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	0.01	0.50
AF35107	7.79	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
AG00000	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	1.00	1.00	6.70	1.00	2.64	1.00
AG00000	1.67	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	17.03	1.00	20.46	1.00
AG00000	1.57	1.50	1.62	1.00	1.00	1.00	1.00	3.75	1.00	1.00	1.00	6.65	1.00	1.00	99.99
AD00507	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00	1.00
AD00507	0.22	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00	1.00
AD00508	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00	1.00
AD00508	1.33	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00	1.00
AD00508	2.12	1.00	1.00	1.00	1.00	1.00	2.07	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
test comparison totals:															
micro	1.57	1.50	1.62	1.00	1.00	1.00	1.00	3.75	1.00	1.00	1.00	6.65	1.00	1.00	99.99
1 iter	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.50	1.00	1.00	1.00	5.10	1.00	1.30	1.00
>1 iter	1.52	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	12.94	1.00	4.64	1.00
diff.	6.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Table E-15. CEEXEC Comparison Output File with VMS Statistics (Continued)

VAX Ada vs ECSPD JOVIAL(opt on)				VMS SHOW STATUS STATISTICS		
I/O-UNIT	NUMBER	WALL-CLOCK	CPU-TIME	BUFIO	DIRIO	PAGE-FAULTS
7	1	6.01	4.12	57	34	736
8	1	7.68	5.77	59	34	792
7	2	23.81	18.19	155	93	1896
8	2	32.35	26.87	163	93	2580

Table E-15. CEEXEC Comparison Output File with VMS Statistics (Concluded)

VAX Ada vs ECSPD JOVIAL (opt on)			VMS SHOW STATUS COMPARISONS: UNIT 01ADA			1 / UNIT 7 (JOVIAL 1)
NUMBER	MALL-CLOCK	CPU-TIME	BUFIS	DIRIO	PAGE-FAULTS	
1	1.28	1.40	1.04	1.00	1.00	
2	1.36	1.48	1.05	1.00	1.36	

APPENDIX F

TEST DESCRIPTIONS

This appendix describes the function of every ACPS test. These test descriptions are contained in Table F-1 and are ordered according to the ACPS test name sorting conventions described in Section E.3. Many entries in the TEST NAME field of Table F-1 are followed by a comma and an alphabetic character (C, I, M, or S) to indicate that an additional version of the test exists. These additional test names contain this character as the third character in the test name. For example, the test description entry AF03501, S corresponds to two tests: AF03501 and AFS3501. For some tests (e.g., AFD520K) an asterisk is placed to the right of the test name. This signifies that the test is VAX specific, and the test program is not placed on the ACPS ANSI formatted delivery tape. The VERSIONS field of Table F-1 lists any other versions of a test based on test type. For example, the following versions of test AF03500 exist within the ACPS: CF03500, EF03500, SF03500, TF03500, JF03500, FF03500. The test description field in Table F-1 describes what is being measured in each test. The descriptions were derived from commentary prologue contained in each test file. The descriptions were condensed, and corrected for grammatical and descriptive errors. However, these corrections were not made to the corresponding test file prologues.

Table F-1 ACPS Test Descriptions

TEST NAME	VERSIONS	DESCRIPTION
AA00000,M	C/E/S/T/J/F	establishes the run time system dependent statistic overhead of support software(oursys,ourspc,ourtyp,ourdmp). The test should be the first test run in any test sequence.
AF03500	C/E/S/T/J/F	test loop overhead for tests: AF03501,AF03502,AF03503,AF03504,AF03509,AF03510, AF05311,AF05312,AF05313,AF05314,AF05317,AF05318.
AF03501,S	C/E/S/T/J	allocation/deallocation of 10 integer variables declared in a procedure.
AF03502,S	C/E/S/T/J/F	allocation/deallocation/initialization to 0 of 10 integer variables declared in a procedure.
AF03503,S	C/E/S/T/J/F	allocation/deallocation/initialization to non-zero constants of 10 integer variables declared in a procedure.
AF03504,S	C/E/S/T/J/F	allocation/deallocation/initialization to global variables of 10 integer variables declared in a procedure.
AF03509,S	C/E/S/T	allocation/deallocation of 10 integer variables declared in a block statement.
AF03510,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 integer variables declared in a block statement.
AF03511,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 integer variables declared in a block statement.
AF03512,S	C/E/S/T	allocation/deallocation/initialization to global variables of 10 integer variables declared in a block statement.
AF03513,S	C/E/S/T/J	allocation/deallocation of 10 string variables declared in a procedure.
AF03514,S	C/E/S/T/J/F	allocation/deallocation/initialization of 10 string variables declared in a procedure.
AF03517,S	C/E/S/T	allocation/deallocation of 10 string variables declared in a block statement.
AF03518,S	C/E/S/T	allocation/deallocation/initialization to a constant of 10 string variables declared in a block statement.
AF03550	C/E/S/T/J	execution of PRED,SUCC functions for enumeration type MONTHS in 10 statements of the form: TMPESi := months'X(TMPESj); where X is PRED or SUCC.
AFD3551	C/E/S/T/J	execution of PRED,SUCC functions for enumeration type REP_MONTHS in 10 statements of the form: TMPERi := rep_months'X(TMPERj); where X is PRED or SUCC.
AF03600	C/E/S/T/J/F	test loop overhead for tests: AF03601,AF03602,AF03603,AF03604,AF03605,AF03606, AF03607,AF03608,AF03609,AF03610, AF03611,AF03612,AF03613,AF03614,AF03615,AF03616, AF03617,AF03618,AF03619.
AF03601,S	C/E/S/T/J	allocation/deallocation of 10 1 dimensional(10) integer arrays declared in a procedure.
AF03602,S	C/E/S/T/J/F	allocation/deallocation/initialization to 0 of 10 1 dimensional(10) integer arrays declared in a procedure.
AF03603,S	C/E/S/T/J/F	allocation/deallocation/initialization to non-zero constants of 10 1 dimensional(10) integer arrays declared in a procedure.
AF03604,S	C/E/S/T/J	allocation/deallocation of 10 2 dimensional(3,3) integer arrays declared in a procedure.
AF03605,S	C/E/S/T/J/F	allocation/deallocation/initialization to 0 of 10 2 dimensional(3,3) integer arrays declared in a procedure.
AF03606,S	C/E/S/T/J/F	allocation/deallocation/initialization to non-zero constants of 10 2 dimensional(3,3) integer arrays declared in a procedure.
AF03607,S	C/E/S/T/J	allocation/deallocation of 10 3 dimensional(3,3,3) integer arrays declared in a procedure.
AF03608,S	C/E/S/T/J/F	allocation/deallocation/initialization to 0 of 10 3 dimensional(3,3,3) integer arrays declared in a procedure.
AF03609,S	C/E/S/T/J/F	allocation/deallocation/initialization to non-zero

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AF03610,S	C/E/S/T/J	constants of 10 3 dimensional(3,3,3) integer arrays declared in a procedure. allocation/deallocation of 10 3 dimensional(3,3,3) integer arrays declared in a procedure. The variables are not referenced and should not be allocated (compare to AF03607).
AF03611,S	C/E/S/T	allocation/deallocation of 10 1 dimensional(10) integer arrays declared in a block statement.
AF03612,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 1 dimensional(10) integer arrays declared in a block statement.
AF03613,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 1 dimensional(10) integer arrays declared in a block statement.
AF03614,S	C/E/S/T	allocation/deallocation of 10 2 dimensional(3,3) integer arrays declared in a block statement.
AF03615,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 2 dimensional(3,3) integer arrays declared in a block statement.
AF03616,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 2 dimensional(3,3) integer arrays declared in a block statement.
AF03617,S	C/E/S/T	allocation/deallocation of 10 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03618,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03619,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03630	C/E/S/T	test loop overhead for tests: AF03631,AF03632,AF03633,AF03634,AF03635,AF03636, AF03637,AF03638,AF03639,AF03641,AF03642,AF03643,AF03644,AF03645,AF03646,AF03647,AF03648,AF03649.
AF03631,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 1 dimensional(10) integer arrays declared in a procedure.
AF03632,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 1 dimensional(10) integer arrays declared in a procedure.
AF03633,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 1 dimensional(10) integer arrays declared in a procedure.
AF03634,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a procedure.
AF03635,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a procedure.
AF03636,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a procedure.
AF03637,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a procedure.
AF03638,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a procedure.
AF03639,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a procedure.
AF03641,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 1 dimensional(10) integer arrays declared in a block statement.
AF03642,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 1 dimensional(10) integer arrays declared in a block statement.

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF03643,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 1 dimensional(10) integer arrays declared in a block statement.
AF03644,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a block statement.
AF03645,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a block statement.
AF03646,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 2 dimensional(3,3) integer arrays declared in a block statement.
AF03647,S	C/E/S/T	allocation/deallocation of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03648,S	C/E/S/T	allocation/deallocation/initialization to 0 of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03649,S	C/E/S/T	allocation/deallocation/initialization to non-zero constants of 10 dynamically sized 3 dimensional(3,3,3) integer arrays declared in a block statement.
AF03700	C/E/S/T/J	test loop overhead for tests: af03701,AF03702,AF03703.
AF03701,S	C/E/S/T/J	allocation/deallocation of 10 1 dimensional(10) arrays with discriminated record components(type VARIANT_RECORDS) declared in a procedure. A variable is used as the record discriminant.
AF03702,S	C/E/S/T/J	allocation/deallocation of 10 1 dimensional(10) arrays with discriminated record components(type VARIANT_RECORDS) declared in a procedure. A constant is used as the record discriminant.
AF03703,S	C/E/S/T/J	allocation/deallocation/initialization to a record aggregate of 10 1 dimensional(10) arrays with discriminated record components(type VARIANT_RECORDS) declared in a procedure. A constant is used as the record discriminant.
AF03800	C/E/S/T/J	test loop overhead for tests: AF03801,AF03802,AF03803, AF03804.
AF03801,S	C/E/S/T/J	allocation/deallocation of 10 1 dimensional(10) arrays with record access components declared in a procedure. The access type(RECORD_POINTER) is globally defined in package ourspc.
AF03802,S	C/E/S/T/J	allocation/deallocation/initialization to a record aggregate of 10 1 dimensional(10) arrays with record access components declared in a procedure. The access type(RECORD_POINTER) is globally defined in package ourspc.
AF03803,S	C/E/S/T/J	allocation/deallocation of 10 1 dimensional(10) arrays with record access components declared in a procedure. The access type used is declared within the procedure and the test includes the time to create/remove the collection of access objects associated with the access type.
AF03804,S	C/E/S/T/J	allocation/deallocation/initialization to a record aggregate of 10 1 dimensional(10) arrays with record access components declared in a procedure. The access type used is declared within the procedure and the test includes the time to create/remove the collection of access objects associated with the access type.
AF04120	C/E/S/T/J/F	3 assignment statements using integer array slices with constant slice ranges.
AF04121	C/E/S/T/J/F	3 assignment statements using integer array slices with scalar variable slice ranges. Compare to AF04120.
AF04122	C/E/S/T/J/F	3 assignment statements using character string slices with constant slice ranges.
AF04123	C/E/S/T/J/F	3 assignment statements using character string slices

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF04124	C/E/S/T/J/F	with scalar variable slice ranges. Compare to AF04122.
AF04125	C/E/S/T/J/F	3 assignment statements of constant aggregates to integer array slices with constant slice ranges.
AF04126	C/E/S/T/J/F	3 assignment statements of constant aggregates to integer array slices with variable slice ranges. Compare to AF04124.
AF04127	C/E/S/T/J/F	3 assignment statements of constant aggregates to slices of string variables with constant slice indices.
AFD4128	C/E/S/T/J	3 assignment statements of constant aggregates to slices of string variables using variable slice and aggregate named association ranges. Compare to AF04126.
AF04129	C/E/S/T/J/F	3 assignment statements of constant aggregates to slices of packed string variables using non-constant slice and aggregate named association ranges. Compare to AF04127.
AFD412A	C/E/S/T/J	3 assignment statements using boolean array slices with constant slice ranges.
AF0412B	C/E/S/T/J	3 assignment statements using record array slices with constant slice ranges.
AFD412C	C/E/S/T/J	3 assignment statements using packed record(type PACKED_RECORDS) array slices with constant slice ranges. Compare to AF0412B.
AFD412D	C/E/S/T/J	3 assignment statements using packed record(type REP_RECORDS) array slices with constant slice ranges. Compare to AF0412B, AFD412C.
AF04130	C/E/S/T/J	references to integer components of records (type RECORDS) in 10 statements of the form: TMPRCi.COMP_I1 := TMPRCj.COMP_I1;.
AF04131	C/E/S/T/J	references to integer components of records (type RECORDS) in 10 statements of the form: TMPRCi.COMP_I2 := TMPRCj.COMP_I2;.
AF04132	C/E/S/T/J	references to boolean components of records (type RECORDS) in 10 statements of the form: TMPRCi.COMP_B1 := TMPRCj.COMP_B1;.
AF04133	C/E/S/T/J	references to enumeration components of records (type RECORDS) in 10 statements of the form: TMPRCi.COMP_E := TMPRCj.COMP_E;.
AFD4135	C/E/S/T/J	references to integer components of packed records (type PACKED_RECORDS) in 10 statements of the form: TMPRPi.COMP_I1 := TMPRPj.COMP_I1;. Compare to AF04130.
AFD4136	C/E/S/T/J	references to integer components of packed records (type PACKED_RECORDS) in 10 statements of the form: TMPRPi.COMP_I2 := TMPRPj.COMP_I2;. Compare to AF04131.
AFD4137	C/E/S/T/J	references to boolean components of packed records (type PACKED_RECORDS) in 10 statements of the form: TMPRPi.COMP_B1 := TMPRPj.COMP_B1;. Compare to AF04132.
AFD4138	C/E/S/T/J	references to enumeration components of packed records (type PACKED_RECORDS) in 10 statements of the form: TMPRPi.COMP_E := TMPRPj.COMP_E;. Compare to AF04133.
AFD4139	C/E/S/T/J	references to integer components of records packed by rep.specs. (type REP_RECORDS, bits 0..4) in 10 statements of the form: TMPRRi.COMP_I1 := TMPRRj.COMP_I1;. Compare to AFD4135.
AFD413A	C/E/S/T/J	references to integer components of records packed by rep.specs. (type REP_RECORDS, bits 5..9) in 10 statements of the form: TMPRRi.COMP_I2 := TMPRRj.COMP_I2;. Compare to AFD4136.
AFD413B	C/E/S/T/J	references to boolean components of records packed by rep.specs. (type REP_RECORDS, bit 10) in 10 statements of the form: TMPRRi.COMP_B1 := TMPRRj.COMP_B1;. Compare to AFD4137.
AFD413C	C/E/S/T/J	references to enumeration components of records packed by rep.specs. (type REP_RECORDS, bits 12..15) in 10 statements of the form: TMPRRi.COMP_E :=

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF0413D	C/E/S/T/J	TMPPRj.COMP_E; Compare to AFD4138. references to integer components of unconstrained record variants (type VARIANT_RECORDS) in 10 statements of the form: TMPVSi.COMP_I1 := TMPVSj.COMP_I1;.
AFD413E	C/E/S/T/J	references to integer components of packed unconstrained record variants (type PACKED_VARIANT) in 10 statements of the form: TMPVPi.COMP_I1 := TMPVPj.COMP_I1; Compare to AF0413D.
AFD413G	C/E/S/T/J	references to packed string components of records (type RECORD_TYPE) in 10 statements of the form: TMPPRi.String_Comp := TMPPRj.String_Comp;.
AF0413H	C/E/S/T/J	references to access type components of record access objects (type RECORD_POINTER) in 10 statements of the form: TMPPPi.Pointer_Comp := TMPPPj.Pointer_Comp;.
AF0413I	C/E/S/T/J	references to integer components of records (type RECORD_TYPE) in 10 statements of the form: TMPPRi.Int_Comp := TMPPRj.Int_Comp;.
AF0413J	C/E/S/T/J	references to integer components of record access objects (type RECORD_POINTER) in 10 statements of the form: TMPPPi.Int_Comp := TMPPPj.Int_Comp; Compare to AF0413I.
AF0413K	C/E/S/T/J	references to integer components of record access objects (type RECORD_POINTER) in 10 statements of the form: TMPPPi.Pointer_Comp.Int_Comp := TMPPPj.Pointer_Comp.Int_Comp; Compare to AF0413J, AF0413I.
AF04310	C/E/S/T/J	10 statements assigning a record aggregate to a record variable (type RECORDS).
AF04311	C/E/S/T/J	10 statements assigning a record aggregate to an unconstrained variant record variable (type VARIANT_RECORDS).
AF04312	C/E/S/T/J	10 statements assigning a record aggregate to a record access variable using the qualifier .all (type RECORD_POINTER).
AF04510	C/E/S/T/J/F	9 logical ANDs all true in one IF statement.
AF04511	C/E/S/T/J/F	9 logical ANDs all false in one IF statement. It is permissible if an elapsed time too short error message is generated for this test.
AF04512	C/E/S/T/J/F	9 logical ORs all true in one IF statement. It is permissible if an elapsed time too short error message is generated for this test.
AF04513	C/E/S/T/J/F	9 logical ORs all false in one IF statement.
AF04514	C/E/S/T/J/F	9 logical XORs all false in one IF statement.
AF04515	C/E/S/T/J/F	1 10 element boolean array AND assignment.
AF04516	C/E/S/T/J/F	1 10 element boolean array relational (equality) assignment.
AF04517	C/E/S/T/J/F	1 10 element boolean array relational (>=) assignment.
AF04518	C/E/S/T/J/F	test loop overhead for tests: AF04519 AF0451a AF0451b AF0451c AF0451d AF0451e AF0451f AF0451g AF0451h AF0451i. The overhead includes all logical operations and statements that are not part of the test but only used to construct it.
AF04519,S	C/E/S/T/J/F	10 relational = in one IF statement (all true) with integer operands.
AF0451A,S	C/E/S/T/J/F	10 relational = in one IF statement (all true) with float operands.
AF0451B,S	C/E/S/T/J/F	10 relational /= in one IF statement (all true) with integer operands.
AF0451C,S	C/E/S/T/J/F	10 relational < in one IF statement (all true) with integer operands.
AF0451D,S	C/E/S/T/J/F	10 relational <= in one IF statement (all true) with integer operands.
AF0451E,S	C/E/S/T/J/F	10 relational > in one IF statement (all true) with integer operands.
AF0451F,S	C/E/S/T/J/F	10 relational >= in one IF statement (all true) with integer operands.

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF0451G,S	C/E/S/T/J/F	integer operands. 10 logical negations in one IF statement (all true) with boolean operands.
AF0451H,S	C/E/S/T/F	10 membership in tests in one IF statement (all true) with integer operands.
AF0451I,S	C/E/S/T/F	10 membership not in tests in one IF statement (all true) with integer operands.
AFD451K	C/E/S/T/J/F	test loop overhead for tests: AFD451L AFD451M AFD451N AFD451O AFD451P AFD451Q AFD451R AFD451S AFD451T. The overhead includes all logical operations and statements that are not part of the test but only used to construct it.
AFD451L,S	C/E/S/T/J/F	10 relational = in one IF statement (all true) with short_short_integer operands.
AFD451M,S	C/E/S/T/J/F	10 relational = in one IF statement (all true) with short_integer operands.
AFD451N,S	C/E/S/T/F	10 relational = in one IF statement (all true) with long_float operands.
AFD451O,S	C/E/S/T/F	10 relational = in one IF statement (all true) with long_long_float operands.
AFD451P,S	C/E/S/T/J/F	10 relational < in one IF statement (all true) with short_short_integer operands.
AFD451Q,S	C/E/S/T/J/F	10 relational < in one IF statement (all true) with short_integer operands.
AFD451R,S	C/E/S/T/F	10 relational < in one IF statement (all true) with long_float operands.
AFD451S,S	C/E/S/T/F	10 relational < in one IF statement (all true) with long_long_float operands.
AFD451T,S*	C/E/S/T/J/F	10 relational = in one IF statement (all true) with d_float operands.
AF0451V	C/E/S/T/J/F	test loop overhead for tests: AF0451W AF0451X AF0451Y. The overhead includes all logical operations and statements that are not part of the test but only used to construct it.
AF0451W,S	C/E/S/T/J/F	10 relational /= in one IF statement (all false) with integer operands.
AF0451X,S	C/E/S/T/J/F	10 relational /= in one IF statement (all false) with float operands.
AF0451Y,S	C/E/S/T/J/F	10 relational < in one IF statement (all false) with integer operands.
AF04520	C/E/S/T/J/F	10 element integer array = test in one IF statement (True). Includes execution of the if + the then part (tmpisl := 100).
AF04521	C/E/S/T/J/F	10 element integer array >= test in one IF statement (True). Includes execution of the if + the then part (tmpisl := 100).
AF04522	C/E/S/T/J/F	10 element integer array = test in one IF statement (False). Includes execution of the if + the else part (tmpis2 := 100).
AFD4523	C/E/S/T/J/F	10 element short_short_integer array = test in one IF statement (true). Includes execution of the if + the then part (tmpisl := 100). Compare to AFD451L.
AFD4524	C/E/S/T/J/F	10 element short_integer array = test in one IF statement (true). Includes execution of the if + the then part (tmpisl := 100). Compare to AFD451M.
AFD4525	C/E/S/T/F	10 element long_float array = test in one IF statement (true). Includes execution of the if + the then part (tmpisl := 100). Compare to AFD451N.
AFD4526	C/E/S/T/F	10 element long_long_float array = test in one IF statement (true). Includes execution of the if + the then part (tmpisl := 100). Compare to AFD451O.
AF04527	C/E/S/T/J	10 element boolean array = test in one IF statement (true). Includes execution of the if + the then part (tmpisl := 100).
AFD4528	C/E/S/T	10 element packed boolean array = test in one IF

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF04529	C/E/S/T/J/F	statement (true). Includes execution of the if + the then part(tmpisl := 100). Compare to AF04527.
AFD452A	C/E/S/T/J	10 character string = test in one IF statement (true). Includes execution of the if + the then part(tmpisl := 100).
AF04530	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 integer additions of scalar variables to a scalar variable.
AF04531	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 integer subtractions of scalar variables from a scalar variable.
AF04532	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 float additions of scalar variables to a scalar variable.
AF04533	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 float subtractions of scalar variables from a scalar variable.
AF04534	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 constant character concatenations to a string variable.
AF04535	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 constant concatenations of strings of length 3 to a string variable.
AF04536	C/E/S/T/J/F	1 assignment statement assigning an expression with 9 integer array element concatenations to an array slice.
AF04537	C/E/S/T/J/F	10 integer increments by 1. Ten assignment statements using scalar variable operands were used to construct the test.
AF04538	C/E/S/T/J/F	10 integer increments by 10. Ten assignment statements using scalar variable operands were used to construct the test.
AF04539	C/E/S/T/J/F	10 integer decrements by 1. Ten assignment statements using scalar variable operands were used to construct the test.
AF0453A	C/E/S/T/J/F	10 integer decrements by 10. Ten assignment statements using scalar variable operands were used to construct the test.
AF0453B	C/E/S/T/J	1 assignment statement assigning an expression with 9 fixed point additions of scalar variables to a scalar variable.
AF0453C	C/E/S/T/J	1 assignment statement assigning an expression with 9 fixed point subtractions of scalar variables from a scalar variable.
AF04540	C/E/S/T/J/F	10 assignment statements assigning a negated integer scalar variable to a scalar variable.
AF04541	C/E/S/T/J/F	10 assignment statements assigning a negated float scalar variable to a scalar variable.
AF04550	C/E/S/T/J/F	3 assignment statements of the form $x := x \bmod y$ where x , y and z are scalar variables and x is 0.
AF04551	C/E/S/T/J/F	3 assignment statements of the form $x := y \bmod z$ where x , y and z are positive scalar variables.
AF04552	C/E/S/T/J/F	3 assignment statements of the form $x := y \bmod z$ where x , y and z are scalar variables and only z is negative.
AF04553	C/E/S/T/J/F	3 assignment statements of the form $x := y \bmod z$ where x , y and z are scalar variables and y is zero.
AF04554	C/E/S/T/J/F	3 assignment statements of the form $x := y \bmod z$ where x , y and z are scalar variables and are >0 .
AF04555	C/E/S/T/J/F	3 assignment statements of the form $x := y \bmod z$ where x , y and z are scalar variables and only z is negative.
AF04556	C/E/S/T/J/F	1 assignment statement assigning an expression with 7 integer multiplies of scalar variables to a scalar variable.
AF04557	C/E/S/T/J/F	1 assignment statement assigning an expression with 7

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		integer divides of scalar variables to a scalar variable.
AF04558	C/E/S/T/J/F	1 assignment statement assigning an expression with 7 float multiplies of scalar variables to a scalar variable.
AF04559	C/E/S/T/J/F	1 assignment statement assigning an expression with 7 float divides of scalar variables to a scalar variable.
AFD455D	C/E/S/T/J/F	3 short_short_integer multiplies in 1 assignment statement using scalar variable operands.
AFD455E	C/E/S/T/J/F	7 short_integer multiplies in 1 assignment statement using scalar variable operands.
AFD455F	C/E/S/T/F	7 long_float multiplies in 1 assignment statement using scalar variable operands.
AFD455G	C/E/S/T/F	7 long_long_float multiplies in 1 assignment statement using scalar variable operands.
AFD455H*	C/E/S/T/J/F	7 d_float multiplies in 1 assignment statement using scalar variable operands.
AFD455I	C/E/S/T/J/F	3 short_short_integer divides in 1 assignment statement using scalar variable operands.
AFD455J	C/E/S/T/J/F	7 short_integer divides in 1 assignment statement using scalar variable operands.
AFD455K	C/E/S/T/F	7 long_float multiplies in 1 assignment statement using scalar variable operands.
AFD455L	C/E/S/T/F	7 long_long_float divides in 1 assignment statement using scalar variable operands.
AFD455M*	C/E/S/T/J/F	7 d_float divides in 1 assignment statement using scalar variable operands.
AF0455N	C/E/S/T/J	1 assignment statement assigning an expression with 7 fixed point multiplies of scalar variables to a scalar variable.
AF0455O	C/E/S/T/J	1 assignment statement assigning an expression with 7 fixed point divides of scalar variables to a scalar variable.
AF04560	C/E/S/T/J/F	6 assignment statements of the form $x := \text{abs}(y)$ where x and y are scalar variables.
AF04562	C/E/S/T/J/F	1 assignment statement assigning an expression with 2 integer exponentiations of scalar variables to a scalar variable.
AF04563	C/E/S/T/J/F	1 assignment statement assigning an expression with 2 float exponentiations of scalar variables to a scalar variable.
AFD4566	C/E/S/T/J	3 short_short_integer exponentiations. Three assignment statements with scalar operands were used to construct the test.
AFD4567	C/E/S/T/J/F	3 short_integer exponentiations. Three assignment statements with scalar operands were used to construct the test.
AFD4568	C/E/S/T/F	3 long_float exponentiations. Three assignment statements with scalar operands were used to construct the test.
AFD4569	C/E/S/T/F	3 long_long_float exponentiations. Three assignment statements with scalar operands were used to construct the test.
AFD456A*	C/E/S/T/J/F	3 d_float exponentiations. Three assignment statements with scalar operands were used to construct the test.
AF04600	C/E/S/T/J/F	10 type conversions (integer to float).
AF04601	C/E/S/T/J/F	10 type conversions (float to integer).
AF04602	C/E/S/T/J/F	10 type conversions of an integer quantity to one with a smaller range(integer to one_to_3).
AFD4603	C/E/S/T/J/F	10 type conversions (short_short_integer to short_integer).
AFD4604	C/E/S/T/J/F	10 type conversions (short_short_integer to integer).
AFD4605	C/E/S/T/J/F	10 type conversions (short_integer to integer).
AFD4606	C/E/S/T/J/F	10 type conversions (integer to short_integer).
AFD4607	C/E/S/T/J/F	10 type conversions (integer to short_short_integer).

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD4608	C/E/S/T/F	10 type conversions (float to long_float).
AFD4609	C/E/S/T/F	10 type conversions (float to long_long_float).
AFD460A	C/E/S/T/F	10 type conversions (long_long_float to long_float).
AFD460B	C/E/S/T/F	10 type conversions (long_long_float to float).
AFD460C*	C/E/S/T	10 type conversions (long_float to d_float).
AFD460D*	C/E/S/T	10 type conversions (d_float to long_float).
AF0460E	C/E/S/T/J	10 type conversions (fixed point to float).
AF0460F	C/E/S/T/J	10 type conversions (float to fixed point).
AF0460G	C/E/S/T/J	10 type conversions (integer to fixed point).
AF0460H	C/E/S/T/J	10 type conversions (fixed point to integer).
AF0460I	C/E/S/T/J/F	10 type conversions (integer to positive).
AF0460J	C/E/S/T/J	10 type conversions (five_bit_integer to integer). The five_bit_integer is contained in a component(comp_il) of a record(type RECORDS).
AFD460K	C/E/S/T/J	10 type conversions (five_bit_integer to integer). The five_bit_integer is contained in a component(comp_il) of a packed record(type PACKED_RECORDS). Compare to AFD460J.
AFD460L	C/E/S/T/J	10 type conversions (five_bit_integer to integer). The five_bit_integer is contained in a component(comp_il) of a packed record(type REP_RECORDS). Compare to AFD460J.
AF0460M	C/E/S/T/J	10 type conversions (integer to five_bit_integer). The five_bit_integer object is contained in a component(comp_il) of a record(type RECORDS).
AFD460N	C/E/S/T/J	10 type conversions (integer to five_bit_integer). The five_bit_integer object is contained in a component(comp_il) of a packed record(type PACKED_RECORDS). Compare to AFD460M.
AFD4600	C/E/S/T/J	10 type conversions (integer to five_bit_integer). The five_bit_integer object is contained in a component(comp_il) of a packed record(type REP_RECORDS). Compare to AFD460M.
AFD4800	C/S/T	test loop overhead for tests: AFD4801, AFD4802, AFD4803. The overhead includes all statements that are not part of the test but only used to construct it.
AFD4801,S	C/S/T	100 statements dynamically allocating/initializing a record object(type RECORDS) and assigning the access value for it to an access variable whose type is locally defined. The storage_size attribute for the access type is determined by the compiler and placed in run statistic 82.
AFD4802,S	C/S/T	test AFD4801 plus 100 statements to use unchecked storage deallocation to reclaim previously allocated storage. Compare to AFD4801.
AFD4803,S	C/S/T	same test as AFD4802 except the storage_size attribute of the access type used is set by the test program to an implementation dependent value.
AF05200	C/E/S/T/J/F	10 integer assignments. The test assigns a scalar variable to a scalar variable.
AF05201	C/E/S/T/J/F	10 integer assignments. A scalar constant is assigned to a scalar variable.
AF05202	C/E/S/T/J/F	10 float assignments. The test assigns a scalar variable to a scalar variable.
AF05203	C/E/S/T/J/F	10 float assignments. A scalar constant is assigned to a scalar variable.
AF05204	C/E/S/T/J/F	10 boolean assignments. The test assigns a scalar variable to a scalar variable.
AF05205	C/E/S/T/J/F	10 boolean assignments. A scalar constant is assigned to a scalar variable.
AF05206	C/E/S/T/J/F	10 relational = assignments using scalar integer operands with the result being true.
AF05207	C/E/S/T/J/F	10 relational = assignments using scalar integer operands with the result being false.
AF05208	C/E/S/T/J/F	10 assignments statements of the form x := not y where x,

Table F-1 ACPs Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AF05209	C/E/S/T/J/F	y are scalar variables. 10 logical AND assignment statements using scalar boolean operands.
AF0520A	C/E/S/T/J/F	10 logical OR assignment statements using scalar boolean operands.
AF0520B	C/E/S/T/J	10 fixed point assignments. The test assigns a scalar variable to a scalar variable.
AFD520C	C/E/S/T/J/F	10 short_short_integer assignments. The test assigns a scalar variable to a scalar variable.
AFD520D	C/E/S/T/J/F	10 short_short_integer assignments. The test assigns a scalar constant to a scalar variable.
AFD520E	C/E/S/T/J/F	10 short_integer assignments. The test assigns a scalar variable to another scalar variable.
AFD520F	C/E/S/T/J/F	10 short_integer assignments. The test assigns a scalar constant to a scalar variable.
AFD520G	C/E/S/T/F	10 long_float assignments. The test assigns a scalar variable to a scalar variable.
AFD520H	C/E/S/T/F	10 long_float assignments. The test assigns a scalar constant to a scalar variable.
AFD520I	C/E/S/T/F	10 long_long_float assignments. The test assigns a scalar variable to a scalar variable.
AFD520J	C/E/S/T/F	10 long_long_float assignments. The test assigns a scalar constant to a scalar variable.
AFD520K*	C/E/S/T/J/F	10 d_float assignments. The test assigns a scalar variable to a scalar variable.
AFD520L*	C/E/S/T/J/F	10 d_float assignments. The test assigns a scalar constant to a scalar variable.
AF0520M	C/E/S/T/J	10 enumeration assignments. The test assigns a scalar variable to a scalar variable.
AFD520N	C/E/S/T/J	10 enumeration assignments. The test assigns a scalar variable to a scalar variable. The variables are of type REP_MONTHS. Compare to AF0520M.
AF0520O	C/E/S/T/J	10 record assignments. The test assigns a record object to a record object.
AFD520P	C/E/S/T/J	10 record assignments. The test assigns a packed record object to a packed record object. The record objects are of type PACKED_RECORDS. Compare to AF0520O.
AFD520Q	C/E/S/T/J	10 record assignments. The test assigns a packed record object to a packed record object. The record objects are of type REP_RECORDS. Compare to AF0520O, AFD520P.
AF0520R	C/E/S/T/J	10 record access variable assignments of the form: TMPPPj := TMPPPj;.
AF0520S	C/E/S/T/J	10 record access variable assignments of the form: TMPPPj.ALL := TMPPPj.ALL;.
AF05210	C/E/S/T/J/F	10 integer array assignments with the subscript a variable. A constant is assigned to an array element.
AF05211	C/E/S/T/J/F	10 integer array assignments with the subscript a constant. A constant is assigned to an array element.
AF05212	C/E/S/T/J/F	10 float array assignments with the subscript a variable. A scalar variable is assigned to an array element.
AF05213	C/E/S/T/J/F	10 float array assignments with the subscript a constant. A scalar variable is assigned to an array element.
AF05214	C/E/S/T/J/F	10 boolean array assignments with the subscript a variable. A scalar variable is assigned to an array element.
AF05215	C/E/S/T/J/F	10 boolean array assignments with the subscript a constant. A scalar variable is assigned to an array element.
AFD5218	C/E/S/T/J/F	10 short_short_integer array assignments with the subscript a scalar variable. A scalar variable is assigned to an array element.
AFD5219	C/E/S/T/J/F	10 short_short_integer array assignments with the subscript a scalar constant. A scalar variable is

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD521A	C/E/S/T/J/F	assigned to an array element. 10 short_integer array assignments with the subscript a scalar variable. A scalar variable is assigned to an array element.
AFD521B	C/E/S/T/J/F	10 short_integer array assignments with the subscript a scalar constant. A scalar variable is assigned to an array element.
AFD521C	C/E/S/T/F	10 long_float array assignments with the subscript a scalar variable. A scalar variable is assigned to an array element.
AFD521D	C/E/S/T/F	10 long_float array assignments with the subscript a scalar constant. A scalar variable is assigned to an array element.
AFD521E	C/E/S/T/F	10 long_long_float array assignments with the subscript a scalar variable. A scalar variable is assigned to an array element.
AFD521F	C/E/S/T/F	10 long_long_float array assignments with the subscript a scalar constant. A scalar variable is assigned to an array element.
AF0521H	C/E/S/T/J/F	5 2-dimensional integer array assignments with the subscript a variable. A scalar variable is assigned to an array element.
AF0521I	C/E/S/T/J/F	5 3-dimensional integer array assignments with the subscript a variable. A scalar variable is assigned to an array element.
AF0521J	C/E/S/T/J/F	10 integer array assignments with the subscript a variable typed with the range the same as the dimensions (compare to AF0521D). A constant is assigned to an array element.
AF0521K	C/E/S/T/J/F	5 2-dimensional integer array assignments with the subscript variables typed with ranges the same as the dimension (compare to AF0521H). A constant is assigned to an array element.
AFD521K	C/E/S/T/J	10 packed boolean array assignments with the subscript a variable. Compare to AF0521J.
AF0521L	C/E/S/T/J	10 character array assignments with the subscript a variable(TMPCSI(TMPISi) := TMPCS4(j);).
AFD521M	C/E/S/T/J	10 packed character array assignments with the subscript a variable(TMPCP1(TMPISi) := TMPCP4(j);). Compare to AF0521L.
AF0521N	C/E/S/T/J	10 record array assignments with the subscript a variable(TMPRAS(TMPISi) := TMPRCj;).
AFD521O	C/E/S/T/J	10 packed record(type PACKED RECORDS) array assignments with the subscript a variable(TMPRAP(TMPISi) := TMPRPj;). Compare to AF0521N.
AFD521P	C/E/S/T/J	10 packed record(type REP RECORDS) array assignments with the subscript a variable(TMPRAR(TMPISi) := TMPRRj;). Compare to AFD521O.
AF05300	C/E/S/T/J/F	test loop overhead for tests: AF05301 AF05302 AF05303. The overhead includes all statements that are not part of the test but only used to construct it.
AF05301,S	C/E/S/T/J/F	10 IF statements with no else part. The if condition fails.
AF05302,S	C/E/S/T/J/F	10 nested IF statements. The if condition fails.
AF05303,S	C/E/S/T/J/F	1 IF and 9 ELSIF statements. The if condition fails.
AF05305	C/E/S/T/J/F	test loop overhead for tests: AF05306 AF05307. The overhead includes all statements that are not part of the test but only used to construct it.
AF05306,S	C/E/S/T/J/F	10 IF statements with the then part executed.
AF05307,S	C/E/S/T/J/F	10 IF statements with the else part executed.
AF05400	C/E/S/T/J/F	test loop overhead for tests: AF05401 AF05402 AF05403 AF05404 AF05405 AF05406 AF05407. The overhead includes statements that are not part of the test but only used to construct it.
AF05401,S	C/E/S/T/J/F	1 case statement with 10 alternatives in increasing

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		sequential numerical order. The last alternative is executed.
AF05402,S	C/E/S/T/J/F	1 case statement with 10 alternatives in decreasing sequential numerical order. The last alternative is executed.
AF05403,S	C/E/S/T/J/F	1 case statement with 10 alternatives in increasing sequential numerical order. The (others) alternative is executed.
AF05404,S	C/E/S/T/J/F	1 case statement with 10 alternatives increasing by 2. The last alternative is executed.
AF05405,S	C/E/S/T/J/F	1 case statement with 10 alternatives with large non-contiguous ranges. The last alternative is executed.
AF05406,S	C/E/S/T/J/F	1 case statement with 10 alternatives with large non-contiguous ranges. The (others) alternative is executed.
AF05407,S	C/E/S/T/J/F	1 case statement with 10 alternatives with large contiguous ranges. The last alternative is executed.
AF05501	C/E/S/T/J/F	test loop overhead for tests: AF05502 AF05503 AF05504 AF05505 AF05506 AF05507 AF05508 AF05509 AF0550a AF0550b. The overhead includes all statements that are not part of the loop control statement.
AF05502,S	C/E/S/T/J/F	10 0-iteration FOR loops(loop control => for i in 1..tmpis0).
AF05503,S	C/E/S/T/J/F	10 0-iteration FOR loops in reverse order(loop control =>for i in reverse 1..tmpis0).
AF05504,S	C/E/S/T/J/F	10 0-iteration while loops(loop control=> while tmpisX <= tmpis0).
AF05506	C/E/S/T/J/F	test loop overhead for tests: AF05507 AF05508 AF05509 AF0550A AF0550B. The overhead includes all statements that are not part of the loop control statement.
AF05507,S	C/E/S/T/J/F	10 1-iteration FOR loops(loop control => for i in 1..tmpisl).
AF05508,S	C/E/S/T/J/F	10 1-iteration FOR loops in reverse order(loop control => for i in reverse 1..tmpisl).
AF05509,S	C/E/S/T/J/F	5 2-level 1-iteration nested FOR loops(loop control => for i in 1..tmpisl).
AF0550A,S	C/E/S/T/J/F	5 2-level 1-iteration nested FOR loops with loop control variable referenced but not used(loop control => for i in 1..tmpisl).
AF0550B,S	C/E/S/T/J/F	1 10 level nested loop with 1 iteration per loop.
AF0550D	C/E/S/T/J/F	references to a 1-dimensional array within a FOR loop using the loop iterative as array index(tmpial(i) :=tmpial(i+1)).
AF0550E	C/E/S/T/J/F	references to a 2-dimensional (3,3) array within a 2-level nested FOR loop using the loop iteratives as array indices (tmpia2(i,j):=tmpia2(j,i)).
AF0550F	C/E/S/T/J/F	references to a 3-dimensional (3,3,3) array within a 3-level nested FOR loop using the loop iteratives as array indices (tmpia3(i,j,k):=tmpia3(k,i,j)).
AF06001	C/E/S/T/J/F	invocation of a local procedure with no arguments.
AF06009	C/E/S/T/J/F	invocation of a local procedure with 1 scalar integer input argument.
AF06010	C/E/S/T/J/F	invocation of a local procedure with 1 scalar integer output argument.
AF06011	C/E/S/T/J/F	invocation of a local procedure with 1 scalar integer input/output argument.
AF06013	C/E/S/T/J/F	loop overhead for subsequent tests for local procedure calls with 10 scalar integer arguments, which are tests AF06014 AF06015 AF06016.
AF06014,S	C/E/S/T/J/F	invocation of a local procedure with 10 scalar integer input arguments.
AF06015,S	C/E/S/T/J/F	invocation of a local procedure with 10 scalar integer output arguments.
AF06016,S	C/E/S/T/J/F	invocation of a local procedure with 10 scalar integer

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF06017	C/E/S/T/J/F	input/output arguments. \$reference in a local procedure to 10 scalar integer input arguments with references to the arguments in 10 statements of the form TMPISi := ARGi.
AF06018	C/E/S/T/J/F	\$reference in a local procedure to 10 scalar integer output arguments with references to the arguments in 10 statements of the form ARGi := TMPISi.
AF06019	C/E/S/T/J/F	\$reference in a local procedure to 10 scalar integer input/output arguments with references to the arguments in 10 statements of the form ARGi := ARGj.
AF06023	C/E/S/T/J/F	loop overhead for subsequent tests for local procedure calls to procedures declared with 10 constrained 1-dimensional integer array arguments, which are tests AF06024 AF06025 AF06026 AF06027 AF06028 AF06029 AF06030 AF06031 AF06032. The overhead includes all statements (including the invocation of a 0 argument local procedure) that are not part of the test but only used to construct it.
AF06024,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array input arguments.
AF06025,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array output arguments.
AF06026,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array input/output arguments.
AF06027,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array input arguments.
AF06028,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array output arguments.
AF06029,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array input/output arguments.
AF06030,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array input arguments.
AF06031,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array output arguments.
AF06032,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array input/output arguments.
AF06043	C/E/S/T/J/F	loop overhead for subsequent tests for local procedure calls declared with 10 unconstrained integer array arguments, which are tests AF06044 AF06045 AF06046 AF06047 AF06048 AF06049 AF06050 AF06051 AF06052. The overhead includes all statements (including the invocation of a 0 argument local procedure) that are not part of the test but only used to construct it.
AF06044,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array input arguments.
AF06045,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array output arguments.
AF06046,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 1-dimensional integer array input/output arguments.
AF06047,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array input arguments.
AF06048,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array output arguments.
AF06049,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 2-dimensional integer array input/output arguments.
AF06050,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array input arguments.
AF06051,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array output arguments.
AF06052,S	C/E/S/T/J/F	invocation of a local procedure with 10 constrained 3-dimensional integer array input/output arguments.
AF06060	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi :=

* run statistic #2 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF06061	C/E/S/T/J/F	ARG(TMPISj). \$reference in a local procedure declared with 10 constrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AF06062	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AF06063	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AF06064	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AF06065	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AF06066	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AF06067	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF0521I).
AF06068	C/E/S/T/J/F	\$reference in a local procedure declared with 10 constrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AF06070	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).
AF06071	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AF06072	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AF06073	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AF06074	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AF06075	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AF06076	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AF06077	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 3-dimensional integer array output

\$ run statistic 02 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF05211).
AF06078	C/E/S/T/J/F	\$reference in a local procedure declared with 10 unconstrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AF06101	C/E/S/T/J/F	external procedure call with 0 arguments.
AF06109	C/E/S/T/J/F	external procedure call with 1 scalar integer input argument.
AF06110	C/E/S/T/J/F	external procedure call with 1 scalar integer output argument.
AF06111	C/E/S/T/J/F	external procedure call with 1 scalar integer input/output argument.
AF06113	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 scalar integer arguments, which are tests AF06114 AF06115 AF06116. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06114,S	C/E/S/T/J/F	call of an external procedure with 10 integer input arguments.
AF06115,S	C/E/S/T/J/F	call of an external procedure with 10 integer output arguments.
AF06116,S	C/E/S/T/J/F	call of an external procedure with 10 integer input/output arguments.
AF06117	C/E/S/T/J/F	\$reference in an external procedure to 10 scalar integer input arguments with references to the arguments in 10 statements of the form TMPISi := ARGi.
AF06118	C/E/S/T/J/F	\$reference in an external procedure to 10 scalar integer output arguments with references to the arguments in 10 statements of the form ARGi := TMPISi.
AF06119	C/E/S/T/J/F	\$reference in an external procedure to 10 scalar integer input/output arguments with references to the arguments in 10 statements of the form ARGi := ARGj.
AF06133	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 constrained 1-dimensional integer array arguments, which are tests AF06134 AF06135 AF06136. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06134,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array input arguments.
AF06135,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array output arguments.
AF06136,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array input/output arguments.
AF06137	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).
AF06138	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AF06139	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AF06143	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 constrained 2-dimensional array arguments, which are

\$ run statistic #2 = size of return statement code

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		tests AF06144 AF06145 AF06146 . The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06144,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array input arguments.
AF06145,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array output arguments.
AF06146,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array input/output arguments.
AF06147	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AF06148	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AF06149	C/E/S/T/J/F	\$reference in an external procedure declared with 10 constrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AF06153	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 constrained 3-dimensional integer array arguments, which are tests AF06154 AF06155 AF06156 . The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06154,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array input arguments.
AF06155,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array output arguments.
AF06156,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array input/output arguments.
AF06157	C/E/S/T/J/F	\$reference in an external procedure with 10 constrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AF06158	C/E/S/T/J/F	\$reference in an external procedure with 10 constrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF0521I).
AF06159	C/E/S/T/J/F	\$reference in an external procedure with 10 constrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AF06163	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 unconstrained 1-dimensional integer array arguments, which are tests AF06164 AF06165 AF06166 . The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06164,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array input arguments.
AF06165,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array output arguments.
AF06166,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 1-dimensional integer array input/output arguments.
AF06167	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).

\$ run statistic 02 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF06168	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AF06169	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AF06173	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 unconstrained 2-dimensional integer array arguments, which are tests AF06174 AF06175 AF06176. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06174,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array input arguments.
AF06175,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array output arguments.
AF06176,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 2-dimensional integer array input/output arguments.
AF06177	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AF06178	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AF06179	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AF06183	C/E/S/T/J/F	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 unconstrained 3-dimensional integer array arguments, which are tests AF06184 AF06185 AF06186. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06184,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array input arguments.
AF06185,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array output arguments.
AF06186,S	C/E/S/T/J/F	call of an external procedure with 10 constrained 3-dimensional integer array input/output arguments.
AF06187	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AF06188	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF0521I).
AF06189	C/E/S/T/J/F	\$reference in an external procedure declared with 10 unconstrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AF06192	C/E/S/T/J	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 enumeration arguments, which are tests AF06193 AF06194 AF06195. The overhead includes all statements (including the invocation of a 0 argument external

\$ run statistic #2 = size of return statement code

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		procedure) that are not part of the test but only used to construct it.
AF06193,S	C/E/S/T/J	call of an external procedure with 10 enumeration input arguments.
AF06194,S	C/E/S/T/J	call of an external procedure with 10 enumeration output arguments.
AF06195,S	C/E/S/T/J	call of an external procedure with 10 enumeration input/output arguments.
AF06196	C/E/S/T/J	reference in an external procedure to 10 enumeration input arguments with references to the arguments in 10 statements of the form $TMPESi := ARGi$.
AF06197	C/E/S/T/J	reference in an external procedure to 10 enumeration output arguments with references to the arguments in 10 statements of the form $ARGi := TMPESi$.
AF06198	C/E/S/T/J	reference in an external procedure to 10 enumeration input/output arguments with references to the arguments in 10 statements of the form $ARGi := ARGj$.
AF0619B	C/E/S/T/J	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 unconstrained variant record arguments, which are tests AF0619C AF0619D AF0619E. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF0619C,S	C/E/S/T/J	call of an external procedure with 10 unconstrained variant record input arguments.
AF0619D,S	C/E/S/T/J	call of an external procedure with 10 unconstrained variant record output arguments.
AF0619E,S	C/E/S/T/J	call of an external procedure with 10 unconstrained variant record input/output arguments.
AF0619F	C/E/S/T/J	reference in an external procedure to 10 unconstrained variant record input arguments with references to the arguments in 10 statements of the form $TMPVSi := ARGi$.
AF0619G	C/E/S/T/J	reference in an external procedure to 10 unconstrained variant record output arguments with references to the arguments in 10 statements of the form $ARGi := TMPVSi$.
AF0619H	C/E/S/T/J	reference in an external procedure to 10 unconstrained variant record input/output arguments with references to the arguments in 10 statements of the form $ARGi := ARGj$.
AF0619K	C/E/S/T/J	loop overhead for subsequent tests for external procedure calls of procedures declared with 10 record access arguments (type RECORD_POINTER), which are tests AF0619L AF0619M AF0619N. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF0619L,S	C/E/S/T/J	call of an external procedure with 10 record access input arguments.
AF0619M,S	C/E/S/T/J	call of an external procedure with 10 record access output arguments.
AF0619N,S	C/E/S/T/J	call of an external procedure with 10 record access input/output arguments.
AF0619O	C/E/S/T/J	reference in an external procedure to 10 record access input arguments with references to the arguments in 10 statements of the form $TMPPPi := ARGi$.
AF0619P	C/E/S/T/J	reference in an external procedure to 10 record access output arguments with references to the arguments in 10 statements of the form $ARGi := TMPPPi$.
AF0619Q	C/E/S/T/J	reference in an external procedure to 10 record access input/output arguments with references to the arguments in 10 statements of the form $ARGi := ARGj$.
AFD6201	C/E/S/T/J	call of a local pragma inline procedure with no arguments.
AFD6209	C/E/S/T/J	call of a local pragma inline procedure with 1 integer

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD6210	C/E/S/T/J	input argument. call of a local pragma inline procedure with 1 integer output argument.
AFD6211	C/E/S/T/J	call of a local pragma inline procedure with 1 integer input/output argument.
AFD6213	C/E/S/T/J	loop overhead for subsequent tests for pragma inline procedure calls of procedures declared with 10 scalar integer arguments, which are tests AFD6214 AFD6215 AFD6216 . The overhead includes all statements(including the invocation of a 0 argument local pragma inline procedure) that are not part of the test but only used to construct it.
AFD6214,S	C/E/S/T/J	call of a local pragma inline procedure with 10 integer input arguments.
AFD6215,S	C/E/S/T/J	call of a local pragma inline procedure with 10 integer output arguments.
AFD6216,S	C/E/S/T/J	call of a local pragma inline procedure with 10 integer input/output arguments.
AFD6217	C/E/S/T/J	*reference in a local pragma inline procedure to 10 integer input arguments with references to the arguments in 10 statements of the form TMPISi := ARGi.
AFD6218	C/E/S/T/J	*reference in a local pragma inline procedure to 10 integer output arguments with references to the arguments in 10 statements of the form ARGi := TMPISi.
AFD6219	C/E/S/T/J	*reference in a local pragma inline procedure to 10 integer input/output arguments with references to the arguments in 10 statements of the form ARGi := ARGj.
AFD6223	C/E/S/T/J	loop overhead for subsequent tests for local pragma inline procedure calls of procedures declared with 10 constrained integer array arguments, which are tests AFD6224 AFD6225 AFD6226 AFD6227 AFD6228 AFD6229 AFD6230 AFD6231 AFD6232 . The overhead includes all statements(including the invocation of a 0 argument local pragma inline procedure) that are not part of the test but only used to construct it.
AFD6224,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array input arguments.
AFD6225,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array output arguments.
AFD6226,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array input/output arguments.
AFD6227,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array input arguments.
AFD6228,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array output arguments.
AFD6229,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array input/output arguments.
AFD6230,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array input arguments.
AFD6231,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array output arguments.
AFD6232,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array input/output arguments.
AFD6243	C/E/S/T/J	loop overhead for subsequent tests for local pragma inline procedure calls of procedures declared with 10 unconstrained integer array arguments, which are tests AFD6244 AFD6245 AFD6246 AFD6247 AFD6248 AFD6249 AFD6250 AFD6251 AFD6252 . The overhead includes all statements(

* run statistic #2 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		including the invocation of a 0 argument local pragma inline procedure) that are not part of the test but only used to construct it.
AFD6244,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array input arguments.
AFD6245,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array output arguments.
AFD6246,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 1-dimensional integer array input/output arguments.
AFD6247,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array input arguments.
AFD6248,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array output arguments.
AFD6249,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 2-dimensional integer array input/output arguments.
AFD6250,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array input arguments.
AFD6251,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array output arguments.
AFD6252,S	C/E/S/T/J	invocation of a local pragma inline procedure with 10 constrained 3-dimensional integer array input/output arguments.
AFD6260	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).
AFD6261	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AFD6262	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AFD6263	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AFD6264	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AFD6265	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AFD6266	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AFD6267	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF0521I).
AFD6268	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 constrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,

\$ run statistic 82 = size of return statement code

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD6270	C/E/S/T	TMPIsj,TMPIsj). \$reference in a local pragma inline procedure declared with 10 unconstrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPIsi := ARG(TMPIsj).
AFD6271	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPIsi) := constant (compare to AF05210).
AFD6272	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPIsi) := ARG(TMPIsj).
AFD6273	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPIsi := ARG(TMPIsj,TMPIsj).
AFD6274	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPIsi,TMPIsi) := constant (compare to AF0521H).
AFD6275	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPIsi,TMPIsi) := ARG(TMPIsj,TMPIsj).
AFD6276	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPIsi := ARG(TMPIsj,TMPIsj,TMPIsj).
AFD6277	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPIsi,TMPIsi,TMPIsi) := constant (compare to AF0521I).
AFD6278	C/E/S/T	\$reference in a local pragma inline procedure declared with 10 unconstrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPIsi,TMPIsi,TMPIsi) := ARG(TMPIsj,TMPIsj,TMPIsj).
AFD6301	C/E/S/T	invocation of an external pragma inline procedure with 0 arguments.
AFD6309	C/E/S/T	invocation of an external pragma inline procedure with 1 scalar integer input argument.
AFD6310	C/E/S/T	invocation of an external pragma inline procedure with 1 scalar integer output argument.
AFD6311	C/E/S/T	invocation of an external pragma inline procedure with 1 scalar integer input/output argument.
AFD6313	C/E/S/T	test loop overhead for external pragma inline procedure calls of procedures declared with 10 scalar integer arguments, which are tests AFD6314 AFD6315 AFD6316. The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6314,S	C/E/S/T	invocation of an external pragma inline procedure with 10 scalar integer input arguments.
AFD6315,S	C/E/S/T	invocation of an external pragma inline procedure with 10 scalar integer output arguments.
AFD6316,S	C/E/S/T	invocation of an external pragma inline procedure with 10 scalar integer input/output arguments.
AFD6317	C/E/S/T	\$reference in an external pragma inline procedure to 10 scalar integer input arguments with references to the arguments in 10 statements of the form TMPIsi := ARGi.
AFD6318	C/E/S/T	\$reference in an external pragma inline procedure to 10

\$ run statistic #2 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD6319	C/E/S/T	scalar integer output arguments with references to the arguments in 10 statements of the form ARGi := TMPISi.
AFD6333	C/E/S/T	\$reference in an external pragma inline procedure to 10 scalar integer input/output arguments with references to the arguments in 10 statements of the form ARGi := ARGj.
AFD6334,S	C/E/S/T	loop overhead for subsequent tests for external pragma inline procedure calls of procedures declared with 10 constrained 1-dimensional integer array arguments, which are tests AFD6334 AFD6335 AFD6336 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6335,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array input arguments.
AFD6336,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array output arguments.
AFD6337	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array input/output arguments.
AFD6338	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 constrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).
AFD6339	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 constrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant (compare to AF05210).
AFD6343	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 constrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AFD6344,S	C/E/S/T	loop overhead for subsequent tests for external pragma inline procedure calls of procedures declared with 10 constrained 2-dimensional integer array arguments, which are tests AFD6344 AFD6345 AFD6346 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6345,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 2-dimensional integer array input arguments.
AFD6346,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 2-dimensional integer array output arguments.
AFD6347	C/E/S/T	call of an external pragma inline procedure with 10 constrained 2-dimensional integer array input/output arguments.
AFD6348	C/E/S/T	\$reference in an external pragma inline procedure with 10 constrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AFD6349	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 constrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AFD6353	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 constrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
		loop overhead for subsequent tests for external pragma inline procedure calls of procedures declared with 10

\$ run statistic 02 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
		constrained 3-dimensional integer array arguments, which are tests AFD6354 AFD6355 AFD6356 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6354,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array input arguments.
AFD6355,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array output arguments.
AFD6356,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array input/output arguments.
AFD6357	C/E/S/T	reference in an external pragma inline procedure declared with 10 constrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AFD6358	C/E/S/T	reference in an external pragma inline procedure declared with 10 constrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant(compare to AF05211).
AFD6359	C/E/S/T	reference in an external pragma inline procedure declared with 10 constrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AFD6363	C/E/S/T	loop overhead for subsequent tests for external pragma inline procedure calls of procedures declared with 10 unconstrained 1-dimensional integer array arguments, which are tests AFD6364 AFD6365 AFD6366 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6364,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array input arguments.
AFD6365,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array output arguments.
AFD6366,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 1-dimensional integer array input/output arguments.
AFD6367	C/E/S/T	reference in an external pragma inline procedure declared with 10 unconstrained 1-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj).
AFD6368	C/E/S/T	reference in an external pragma inline procedure declared with 10 unconstrained 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant(compare to AF05210).
AFD6369	C/E/S/T	reference in an external pragma inline procedure declared with 10 unconstrained 1-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi) := ARG(TMPISj).
AFD6373	C/E/S/T	loop overhead for subsequent tests for external pragma inline procedure calls of procedures declared with 10 unconstrained 2-dimensional integer array arguments, which are tests AFD6374 AFD6375 AFD6376 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6374,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 2-dimensional integer array input arguments.
AFD6375,S	C/E/S/T	call of an external pragma inline procedure with 10

* run statistic #2 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFD6376,S	C/E/S/T	constrained 2-dimensional integer array output arguments. call of an external pragma inline procedure with 10 constrained 2-dimensional integer array input/output arguments.
AFD6377	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 2-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj).
AFD6378	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 2-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi) := constant (compare to AF0521H).
AFD6379	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 2-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi) := ARG(TMPISj,TMPISj).
AFD6383	C/E/S/T	loop overhead for subsequent tests for external pragma inline procedure calls of procedures with 10 unconstrained 3-dimensional integer array arguments, which are tests AFD6384 AFD6385 AFD6386 . The overhead includes all statements (including the invocation of a 0 argument external pragma inline procedure) that are not part of the test but only used to construct it.
AFD6384,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array input arguments.
AFD6385,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array output arguments.
AFD6386,S	C/E/S/T	call of an external pragma inline procedure with 10 constrained 3-dimensional integer array input/output arguments.
AFD6387	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 3-dimensional integer array input arguments to an argument in 10 statements of the form TMPISi := ARG(TMPISj,TMPISj,TMPISj).
AFD6388	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 3-dimensional integer array output arguments to an argument in 5 statements of the form ARG(TMPISi,TMPISi,TMPISi) := constant (compare to AF0521I).
AFD6389	C/E/S/T	\$reference in an external pragma inline procedure declared with 10 unconstrained 3-dimensional integer array input/output arguments to an argument in 10 statements of the form ARG(TMPISi,TMPISi,TMPISi) := ARG(TMPISj,TMPISj,TMPISj).
AF06423	C/E/S/T	loop overhead for subsequent tests for external procedure calls with 10 arguments with default parameter values. The tests are AF06424,AF06425,AF06426, AF06427, AF06428,AF06429,AF0642A. The overhead includes all statements (including the invocation of a 0 argument external procedure) that are not part of the test but only used to construct it.
AF06424,S	C/E/S/T	call of an external procedure declared with 10 integer input arguments with default values. The procedure is called with 10 actual arguments. Compare to AF06114.
AF06425,S	C/E/S/T	call of an external procedure declared with 10 integer input arguments with default values. The procedure is called with 0 actual arguments. Compare to AF06114, AF06424.
AF06426,S	C/E/S/T	call of an external procedure declared with 10 constrained 1-dimensional integer array input arguments

* run statistic #2 = size of return statement code

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF06427,S	C/E/S/T	with default values. The procedure is called with 10 actual arguments. Compare to AF06134.
AF06428,S	C/E/S/T	call of an external procedure declared with 10 constrained 1-dimensional integer array input arguments with default values. The procedure is called with 0 actual arguments. Compare to AF06134,AF06426.
AF06429,S	C/E/S/T	call of an external procedure declared with 10 record input arguments with default values. The procedure is called with 10 actual arguments.
AF0642A	C/E/S/T	call of an external procedure declared with 10 record input arguments with default values. The procedure is called with 0 actual arguments. Compare to AF06428.
AF06802	C/E/S/T/J/F	reference in an external procedure to 10 integer input arguments with references to the arguments in 10 statements of the form TMPISi := ARGi. The arguments are declared with default values. The procedure is called with 0 arguments. Compare to AF06117.
AF06806	C/E/S/T/J/F	global scalar integer variable reference from within a called procedure(compare to AF05200).
AF06808	C/E/S/T/J	local scalar integer variable reference from within a called procedure(compare to AF06802).
AF0680A	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form ARGi := ARGj. The access variable type is globally declared(RECORD_POINTER). Compare to AF0520R.
AF0680C	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form LOCi.all := LOCj.all. The access variable type is globally declared(RECORD_POINTER). Compare to AF0520S.
AF0680E	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form LOCi := LOCj. The access type used is locally declared. The storage_size attribute for the access type is determined by the compiler and placed in run statistic #2. Compare to AF06808.
AFD680G	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form LOCi.all := LOCj.all. The access type used is locally declared. The storage_size attribute for the access type is determined by the compiler and placed in run statistic #2.
AFD680I	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form LOCi := LOCj. The access type used is locally declared. The storage_size attribute for the access type is set by the test program and placed in run statistic #2. Compare to AF0680C.
AF06810	C/E/S/T/J	local record access variable reference from within a called procedure in 10 statements of the form LOCi.all := LOCj.all. The access type used is locally declared. The storage_size attribute for the access type is set by the test program and placed in run statistic #2. Compare to AF0680E.
AF06814	C/E/S/T/J	local scalar integer variable reference - 1 procedure level up (compare to AF06806).
AF06816	C/E/S/T/J/F	local scalar integer variable reference - 2 procedure levels up (compare to AF06806).
AF06818	C/E/S/T/J/F	local 1-dimensional array element references(compare to AF05210).
AF06820	C/E/S/T/J/F	local 2-dimensional array element references(compare to AF0521H).
AF06822	C/E/S/T/J	local 3-dimensional array element references(compare to AF0521I).
		local unconstrained variant record reference from within a called procedure in 10 statements of the form

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF06824	C/E/S/T/J	LOCi.COMP_11 = LOCj.COMP_11. Compare to AF0413D. local constrained variant record reference from within a a called procedure in 10 statements of the form LOCi.COMP_11 = LOCj.COMP_11. Compare to AF06822.
AFN9301	C/E/S/T	declaration/activation of a single low priority task within a block statement. Control is not transferred to the low priority task.
AFN9302	C/E/S/T	declaration/activation of a single high priority task within a block statement. Control is transferred to the high priority task.
AFN9303	C/E/S/T	declaration/activation of 10 low priority tasks within a a block statement. Control is not transferred to the low priority task(compare to AFN9301).
AF09501	C/E/S/T	rendezvous with a 0-argument task entry. Run statistic #2 = storage_size attribute of the called task.
AF09502	C/E/S/T	rendezvous with a 0-argument task entry that is the last 5th) accept alternative in a select statement(compare to AF09501). Run statistic #2 = storage_size attribute of the called task.
AF09503	C/E/S/T	rendezvous with a task entry with 10 integer scalar input arguments. Run statistic #2 = storage_size attribute of the called task.
AF09504	C/E/S/T	rendezvous with a task entry with 10 integer scalar output arguments. Run statistic #2 = storage_size attribute of the called task.
AF09505	C/E/S/T	rendezvous with a task entry with 10 1-dimensional integer array output arguments. Run statistic #2 = storage_size attribute of the called task.
AF09506	C/E/S/T	reference in a task accept statement to 10 integer scalar input arguments with references to the arguments in 10 statements of the form TMPISi := ARGi(compare to AF06117).
AF09507	C/E/S/T	reference in a task accept statement with 10 1-dimensional integer array output arguments to an argument in 10 statements of the form ARG(TMPISi) := constant. Compare to AF06168.
AF09508	C/E/S/T	local scalar integer variable reference from within a task accept statement(compare to AF06806).
AF09509	C/E/S/T	global scalar integer variable reference from within a task accept statement(compare to AF06802,AF09508).
AFN9511	C/E/S/T	queue an entry call to a suspended task,suspend the caller and transfer control to a lower priority task.
AF09600	C/E/S/T	references to the clock function in 10 assignment statements of the form: time_i := clock.
AF09601	C/E/S/T	references to the seconds function in 10 assignment statements of the form: DURATION_i := seconds(TIME_i).
AF09602	C/E/S/T	references to the split procedure in 5 statements of the form: split(TIME_i,TMPISi,TMPISj,TMPISK,DURATION_i).
AF09603	C/E/S/T	references to the time of function in 10 assignment statements of the form: TIME_i := time_of(TMPISi,TMPISj, TMPISK,DURATION_i).
AF09604	C/E/S/T	10 assignment statements that add values of type duration to those of type time(TIME_i := TIME_i + DURATION_i).
AF09605	C/E/S/T	10 assignment statements involving subtraction of values of type time(DURATION_i := TIME_i - TIME_0).
AF09606	C/E/S/T	10 boolean assignments involving values of type time in statements of the following form: TMPBSi := TIME_i < TIME_j.
AF09607	C/E/S/T	10 conversions of type duration to type float in 10 statements of the form: TMPRSi := our_float(DURATION_i).
AFN9611	C/E/S/T	wake up from a delay statement with no other tasks executing (test_time = time_awaken - time_delay_expires)
AFN9612	C/E/S/T	wake up from a delay statement with 5 low priority tasks executing in an interleaved fashion through use of

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFN9613	C/E/S/T	delay statements (test_time = time_awaken - time_delay_expires). transfer control to a low priority task not in a wait state after execution of a delay statement in a high priority task. The test time includes the time to execute the delay statement.
AF09710	C/E/S/T	execution of a selective wait statement with 4 select alternatives (accept statements) and an else part. The else part is executed. Run statistic #2 = storage_size attribute of the test task.
AF09712	C/E/S/T	execution of a selective wait statement with 4 select alternatives (accept statements) and an else part. Each accept statement is guarded with a test of the count attribute of an entry queue. The else part is executed. Run statistic #2 = storage_size attribute of the test task.
AF09720	C/E/S/T	conditional entry call, call immediately taken. Compare to AF09501. Run statistic #2 = storage_size attribute of the called task.
AF09721	C/E/S/T	conditional entry call, else part(null) taken. Run statistic #2 = storage_size attribute of the called task.
AF09731	C/E/S/T	timed entry call, call immediately taken. Compare to AF09501, AF09720. Run statistic #2 = storage_size attribute of the called task.
AF09901	C/E/S/T	references to the callable attribute of tasks in 10 assignment statements of the form: TMPBSi := tasks(i) 'callable.
AF09902	C/E/S/T	references to the terminated attribute of tasks in 10 assignment statements of the form: TMPBSi := tasks(i) 'terminated.
AF09903	C/E/S/T	references to the count attribute of task entries in 10 assignment statements of the form: TMPBSi := entry_i'count.
AFM9A01	C/E/S/T	1 abort statement issued from within a block statement aborting a single task created within the block. The task is waiting at an accept statement.
AFM9A02	C/E/S/T	1 abort statement issued from within a block statement aborting 10 tasks created within the block. The tasks are waiting at accept statements. Compare to AFM9A01.
AFM9A03	C/E/S/T	execution of 1 abort statement from within a block and exit from the block. The abort statement aborts 10 tasks waiting at accept statements. Compare to AFM9A02.
AF09B01	C/E/S/T	shared global scalar integer variable reference from within a task accept statement (compare to AF09509).
AFD9C00	C/E/S/T	execution of a work-load from a main program with no subordinate tasks and without a priority pragma.
AFD9C01	C/E/S/T	execution of a work-load by a single subordinate task. Compare to AFD9C00 to see the effect of task scheduling overhead. (run statistic #1 = storage_size attribute of the subordinate task type).
AFD9C02,1	C/E/S/T	execution of a work-load by 5 subordinate tasks. Compare to AFD9C00, AFD9C01 to see the effect of task scheduling overhead (run statistics 1-5 in AFI9C02 contain the task iteration counts in the order the tasks were created).
AFD9C03,1	C/E/S/T	execution of a work-load by 5 subordinate tasks with a forced time-slice interval (VAX=.01secs). Compare to AFD9C02 to see the effect of forced time-slicing (run statistics 1-5 in AFI9C03 contain the task iteration counts in the order the tasks were created).
AFD9C04,1	C/E/S/T	execution of a work-load by 20 subordinate tasks with a forced time-slice interval (VAX=.01secs). Compare to AFD9C03 to see the effect of task-scheduling overhead. (run statistics 1-4 in AFI9C04 contain the total, maximum, minimum and median task iteration counts).

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFMB001	C/S/T	raise/deliver a user defined exception and test for it with the exception choice others.
AFMB003	C/S/T	raise/deliver a user defined exception and test for 5 predefined exceptions and then for the user defined exception (compare to AFMB001).
AFMB005	C/S/T	raise/deliver a numeric error pre-defined exception(divide by zero: $tmpis1:=tmpis2/tmpis0$).
AFMB007	S/T	raise/deliver a numeric range constraint error exception ($tmpil:=tmpis0$).
AFMB009	S/T	raise/deliver an index range constraint error exception($tmpial(tmpis0) := 0$).
AFMB012	C/S/T	raise/deliver a user defined exception which is automatically propagated 1 procedure level upward(propagation time=AFMB012-AFMB001).
AFMB016	C/S/T	raise/deliver a user defined exception which is automatically propagated 2 procedure levels upward(propagation time=AFMB016-AFMB001).
AFMB020	C/S/T	raise/deliver a user defined exception. Use raise statements to propagate the exception to a handler 2 procedure levels upward(compare to AFMB016).
AFMB024	S/T	detect/deliver a numeric error exception(divide by 0) which is automatically propagated 2 procedure levels upward (propagation time=AFMB024-AFMB005. Compare to AFMB016).
AFMB028	S/T	detect/deliver a numeric error exception(divide by 0). Use raise statements to propagate the exception to a handler 2 procedure levels upward(compare to AFMB024).
AFMB032	C/S/T	raise/deliver a user defined exception which is automatically propagated 2 procedure levels upward(propagation time=AFMB032-AFMB001). Each procedure has 10 scalar output arguments(compare to AFMB016).
AFMB036	C/S/T	raise/deliver a user defined exception. Use raise statements to propagate the exception to a handler 2 procedure levels upward. Each procedure has 10 scalar output arguments(compare to AFMB020, AFMB032).
AFMB040	S/T	detect/deliver a numeric error exception(divide by 0) which is automatically propagated 2 procedure levels upward (propagation time=AFMB040-AFMB005). Each procedure has 10 scalar output arguments(compare to AFMB024, AFMB032).
AFMB044	C/S/T	detect/deliver a numeric error exception(divide by 0). Use raise statements to propagate the exception to a handler 2 procedure levels upward. Each procedure has 10 scalar output arguments(compare to AFMB028, AFMB040).
AF0C300	C/E/S/T	non-generic version of test AF0C301 with which its compared (AFCC301 = AF0C301/AF0C300).
AF0C301.C	C/E/S/T	tests execution of a procedure derived from instantiation of a generic package containing a procedure whose compiled code should be the same for each instantiation (only instantiate for unconstrained floating point vectors).
AF0C302	C/E/S/T	same test as AF0C301, except the generic package is instantiated with 3 different numeric types(integer, float, boolean). All three instantiations are referenced. Only the floating point procedure is timed. Compare to AF0C301 to see if memory requirements are greater (i.e. are separate copies of the generic procedure generated for each instantiation?).
AF0C303	C/E/S/T	non-generic version of test AF0C304 with which its compared (AFCC304 = AF0C304/AF0C303).
AF0C304.C	C/E/S/T	tests execution of a function derived from instantiation of a generic package containing vector functions(+,-,*, /) whose compiled code should be different for each instantiation (only instantiate for unconstrained floating point vectors). Only the + function is

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AF0C305	C/E/S/T	referenced or executed. same test as AF0C304, except all functions in the instantiated package are referenced but only the + function is timed. Compare to AF0C304 to see if the memory requirements are greater in this test(i.e. are unreferenced instantiated functions included in the load module).
AF0C306	C/E/S/T	same test as AF0C305, except the generic package is instantiated for 2 datatypes (integer, floating point). Only the floating point instantiation is referenced or timed. Compare memory requirements to AF0C305(i.e. are unreferenced instantiations included in the load module)
AF0C307	C/E/S/T	same test as AF0C306, except all package instantiations are referenced. Compare to AF0C306, AF0C305(note: execution time and memory requirements will differ depending on whether or not a new copy of a function is generated for each instantiation).
AFDC310	C/E/S/T	non-generic version of test AFDC311 with which its compared (AFCC311 = AFDC311/AFDC310).
AFDC311,C	C/E/S/T	tests execution of an inline procedure derived from instantiation of a generic package containing an inline procedure whose compiled code should be the same for each instantiation (only instantiate for unconstrained floating point vectors). Compare to AF0C301.
AFDC313	C/E/S/T	non-generic version of test AFDC314 with which its compared (AFCC314 = AFDC314/AFDC313).
AFDC314,C	C/E/S/T	tests execution of an inline function derived from instantiation of a generic package containing inline vector functions(+,-,*,/) whose compiled code should be different for each instantiation (only instantiate for unconstrained floating point vectors). Only the + function is referenced or executed. Compare to AF0C304.
AFDD600	C/E/S/T/J	3 statements changing representation of unpacked boolean arrays (1..10) to packed boolean arrays.
AFDD601	C/E/S/T/J	3 statements changing representation of packed boolean arrays (1..10) to unpacked boolean arrays.
AFDD602	C/E/S/T/J	10 statements changing representation from objects of enumeration type REP_MONTHS to values of type MONTHS.
AFDD603	C/E/S/T/J	10 statements changing representation from objects of enumeration type MONTHS to values of type REP_MONTHS.
AFDD604	C/E/S/T/J	10 statements changing representation of packed records(type PACK_RECORDS) to unpacked records(type RECORDS).
AFDD605	C/E/S/T/J	10 statements changing representation of unpacked records(type RECORDS) to packed records(type PACK_RECORDS).
AFDD606	C/E/S/T/J	10 statements changing representation of unpacked records(type RECORDS) to packed records(type REP_RECORDS).
AFDD607	C/E/S/T/J	10 statements changing representation of packed records(type REP_RECORDS) to unpacked records(type RECORDS).
AFDD608	C/E/S/T/J	10 statements changing representation of packed records(type REP_RECORDS) to packed records(type PACK_RECORDS).
AFDD609	C/E/S/T/J	10 statements changing representation of packed records(type PACK_RECORDS) to packed records(type REP_RECORDS).
AFDD610	C/E/S/T/J	10 statements changing representation of unpacked record variants(type VARIANT_RECORDS) to packed variant records (type PACKED_VARIANT).
AFDD611	C/E/S/T/J	10 statements changing representation of packed record variants(type PACKED_VARIANT) to unpacked record variants (type VARIANT_RECORDS).
AF0D720	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in package OURSYS. Run statistic 01 = bit size of objects of type integer. Run statistic 02 = bit size of objects which are 1-dimensional(10) integer arrays. Run statistic 03 =

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFDD721	S/T/J/F	bit size of objects which are 2-dimensional(3,3) integer arrays. Run statistic #4 = bit size of objects which are 3-dimensional(3,3,3) integer arrays. Run statistic #5 = bit size of objects of type positive. displays in run statistic fields the size of objects used in tests and defined in package OURSYS. Run statistic #1 = bit size of objects of type float. Run statistic #2 = bit size of objects which are 1-dimensional(10) float arrays. Run statistic #3 = bit size of objects which are 2-dimensional(3,3) float arrays. Run statistic #4 = bit size of objects which are 3-dimensional(3,3,3) float arrays.
AFDD722	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in package OURTYP. Run statistic #1 = bit size of objects of type short_integer. Run statistic #2 = bit size of objects which are 1-dimensional(10) short_integer arrays. Run statistic #3 = bit size of objects which are 2-dimensional(10,10) short_integer arrays.
AFDD723	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in package OURTYP. Run statistic #1 = bit size of objects of type short_short_integer. Run statistic #2 = bit size of objects which are 1-dimensional(10) short_short_integer arrays. Run statistic #3 = bit size of objects which are 2-dimensional(10,10) short_short_integer arrays.
AFDD724	S/T/F	displays in run statistic fields the size of objects used in tests and defined in package OURTYP. Run statistic #1 = bit size of objects of type long_float. Run statistic #2 = bit size of objects which are 1-dimensional(10) long_float arrays. Run statistic #3 = bit size of objects which are 2-dimensional(10,10) long_float arrays.
AFDD725	S/T/F	displays in run statistic fields the size of objects used in tests and defined in package OURTYP. Run statistic #1 = bit size of objects of type long_long_float. Run statistic #2 = bit size of objects which are 1-dimensional(10) long_long_float arrays. Run statistic #3 = bit size of objects which are 2-dimensional(10,10) long_long_float arrays.
AFDD726x	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in package OURTYP. Run statistic #1 = bit size of objects of type d_float. Run statistic #2 = bit size of objects which are 1-dimensional(10) d_float arrays. Run statistic #3 = bit size of objects which are 2-dimensional(10,10) d_float arrays.
AFDD727	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in package OURSYS. Run statistic #1 = bit size of objects of type boolean. Run statistic #2 = bit size of objects which are 1-dimensional(1..10) boolean arrays. Run statistic #3 = bit size of objects which are 2-dimensional(3,3) boolean arrays. Run statistic #4 = bit size of objects which are 3-dimensional(3,3,3) boolean arrays.
AFDD728	S/T/J/F	displays in run statistic fields the size of objects used in tests and defined in packages OURSYS, OURSPC. Run statistic #1 = bit size of objects which are 10 character strings. Run statistic #2 = bit size of objects which are 20 character strings. Run statistic #3 = bit size of objects which are 30 character strings. Run statistic #4 = bit size of objects of type MONTHS.
AFDD729	S/T/J	displays in run statistic fields the size of objects used in tests and defined in package OURSPC. Run

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFDD72A	S/T/J	<p>statistic #1 = bit size of objects which are records of type RECORDS. Run statistic #2 = bit size of objects which are 1-dimensional arrays whose components are of type RECORDS. Run statistic #3 = bit size of objects which are unconstrained variants of the discriminated record type VARIANT_RECORDS. Run statistic #4 = bit size of objects which are records of type RECORD_TYPE. Run statistic #5 = bit size of objects which are access variables of type RECORD_POINTER.</p> <p>displays in run statistic fields the size of objects used in tests and defined in package OURSPC. Run statistic #1 = bit size of objects which are packed records of type PACKED_RECORDS. Compare to run statistic #1 in AFDD729. Run statistic #2 = bit size of objects which are 1-dimensional arrays whose components are of type PACKED_RECORDS. Compare to run statistic #2 in AFDD729. Run statistic #3 = bit size of objects which are unconstrained variants of the discriminated record type PACKED_VARIANT. Compare to run statistic #3 in AFDD729. Run statistic #4 = bit size of objects which are packed 30 character strings. Compare to run statistic #3 in AFDD728. Run statistic #5 = bit size of objects which are packed 1-dimensional(1..10) boolean arrays. Compare to run statistic #2 in AFDD727.</p>
AFDD72B	S/T/J	<p>displays in run statistic fields the size of objects used in tests and defined in package OURSPC. Run statistic #1 = bit size of objects which are packed records of type REP_RECORDS. Compare to run statistic #1 in AFDD72A, AFDD729. Run statistic #2 = bit size of objects which are 1-dimensional arrays whose components are of type REP_RECORDS. Compare to run statistic #2 in AFDD72A, AFDD729. Run statistic #3 = bit size of objects of type REP_MONTHS. Compare to run statistic #4 in AFDD728.</p>
AFDD72C	S/T/J	<p>displays in run statistic fields the size of objects used in tests and defined in package OURSPC. Run statistic #1 = bit size of Pointer Comp component of objects of type RECORD_TYPE. Run statistic #2 = bit size of Enum Comp component of objects of type RECORD_TYPE. Run statistic #3 = bit size of Int Comp component of objects of type RECORD_TYPE. Run statistic #4 = bit size of String_30 component of objects of type RECORD_TYPE. Run statistic #5 = Storage'size attribute for access type RECORD_POINTER.</p>
AFDD72D	S/T/J	<p>displays in run statistic fields the size of components of record objects of type RECORDS used in tests and defined in package OURSPC. Run statistic #1 = bit size of comp_i1 component. Run statistic #2 = bit size of comp_i2 component. Run statistic #3 = bit size of comp_b1 component. Run statistic #4 = bit size of comp_b2 component. Run statistic #5 = bit size of comp_e component.</p>
AFDD72E	S/T/J	<p>displays in run statistic fields the size of components of record objects of type PACKED_RECORDS used in tests and defined in package OURSPC. Run statistic #1 = bit size of comp_i1 component. Run statistic #2 = bit size of comp_i2 component. Run statistic #3 = bit size of comp_b1 component. Run statistic #4 = bit size of comp_b2 component. Run statistic #5 = bit size of comp_e component. Compare to corresponding statistics in AFDD72D.</p>
AFDD72F	S/T/J	<p>displays in run statistic fields the size of components of record objects of type REP_RECORDS used in tests and defined in package OURSPC. Run statistic #1 = bit size of comp_i1 component. Run statistic #2 = bit size of</p>

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AFDD72G	S/T/J	comp_i2 component. Run statistic #3 = bit size of comp_b1 component. Run statistic #4 = bit size of comp_b2 component. Run statistic #5 = bit size of comp_e component. Compare to corresponding statistics in AFDD72D, AFDD72E. displays in run statistic fields the size of components of record objects of type VARIANT RECORDS used in tests and defined in package OURSPC. Run statistic #1 = bit size of comp_i1 component. Run statistic #2 = bit size of comp_i2 component. Run statistic #3 = bit size of comp_b1 component. Run statistic #4 = bit size of comp_b2 component. Run statistic #5 = bit size of comp_e component.
AFDD72H	S/T/J	displays in run statistic fields the size of components of record objects of type PACKED VARIANT used in tests and defined in package OURSPC. Run statistic #1 = bit size of comp_i1 component. Run statistic #2 = bit size of comp_i2 component. Run statistic #3 = bit size of comp_b1 component. Run statistic #4 = bit size of comp_b2 component. Run statistic #5 = bit size of comp_e component. Compare to corresponding statistics in AFDD72G.
AFDD72I	S/T/J	displays in run statistic fields the size of objects used in tests and defined in package OURSYS, OURSPC. Run statistic #1 = bit size of objects which are packed 10 character strings. Run statistic #2 = bit size of objects which are packed 20 character strings. Run statistic #3 = bit size of objects which are packed 30 character strings. Run statistic #4 = bit size of objects of type REP_MONTHS. Compare to corresponding statistics in AFDD728.
AFDDA01	C/E/S/T/J	10 unchecked conversions of floating point operands to integer in 10 statements of the form: TMPISi := float_to_integer(TMPRSsi).
AFDDA02	C/E/S/T/J	10 unchecked conversions of record objects of type REP RECORDS to integer in 10 statements of the form: TMPISi := records_to_integer(TMPRRsi).
AFDE201	C/E/S/T/F	1968 sequential file writes using an operand that is a scalar character variable. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE202	C/E/S/T/F	1968 sequential file writes using an operand that is a 1 element character array. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE203	C/E/S/T/F	240 sequential file writes using an operand that is an 8 element array with character components. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE204	C/E/S/T/F	24 sequential file writes using an operand that is an 82 element array of characters with embedded carriage control. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE205	C/E/S/T/F	1 sequential file write using an operand that is a 1968 element array with character components. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE206	C/E/S/T/F	8040 sequential file writes using an operand that is a scalar character variable. Output to the file is 8040 characters per iteration which fills a standard page of

Table F-1 ACPs Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AFDE207	C/E/S/T/F	computer paper (by default 60 x 132). 8040 sequential file writes using an operand that is a 1 element character array. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE208	C/E/S/T/F	960 sequential file writes using an operand that is an 8 element array with character components. Output to the file is 7800 characters per iteration (including carriage control) which fills a standard page of computer paper (by default 60 x 132).
AFDE209	C/E/S/T/F	60 sequential file writes using an operand that is an 82 element array of characters with embedded carriage control. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE20A	C/E/S/T/F	1 sequential file write using an operand that is a 8040 element array with character components. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE211	C/E/S/T/F	1 sequential file write using an operand that is a 512 element array with integer components. The output to the file consists of 512 integers per iteration.
AFDE212	C/E/S/T/F	512 sequential file writes using an operand that is a scalar integer variable. Output to the file consists of 512 integers per iteration.
AFDE213	C/E/S/T/F	1 sequential file write using an operand that is a 512 element array with floating point components. The output to the file consists of 512 floating point numbers per iteration.
AFDE214	C/E/S/T/F	512 sequential file writes using an operand that is a floating point scalar variable. Output to the file consists of 512 floating point numbers per iteration.
AFDE215	C/E/S/T/F	1 sequential file write using an operand that is a 512 element array with character components. The output to the file consists of 512 characters per iteration.
AFDE216	C/E/S/T/F	512 sequential file writes using an operand that is a scalar character variable. Output to the file consists of 512 characters per iteration.
AFDE231	C/E/S/T/F	1968 sequential file reads using an operand that is a scalar character variable. Input to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE232	C/E/S/T/F	1968 sequential file reads using an operand that is a 1 element character array. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE233	C/E/S/T/F	240 sequential file reads using an operand that is an 8 element array with character components. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE234	C/E/S/T/F	24 sequential file reads using an operand that is an 82 element array of characters with embedded carriage control. Input to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE235	C/E/S/T/F	1 sequential file read using an operand that is a 1968 element array with character components. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE236	C/E/S/T/F	8040 sequential file reads using an operand that is a scalar character variable. Input to the file is 8040

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AFDE237	C/E/S/T/F	characters per iteration which fills a standard page of computer paper (by default 60 x 132). 8040 sequential file reads using an operand that is a 1 element character array. Input from the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE238	C/E/S/T/F	960 sequential file reads using an operand that is an 8 element array with character components. Input from the file is 7800 characters per iteration (including carriage control) which fills a standard page of computer paper (by default 60 x 132).
AFDE239	C/E/S/T/F	60 sequential file reads using an operand that is an 82 element array of characters with embedded carriage control. Input to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE23A	C/E/S/T/F	1 sequential file read using an operand that is a 8040 element array with character components. Input from the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE241	C/E/S/T/F	1 sequential file read using an operand that is a 512 element array with integer components. Input from the file is a block of integers per iteration.
AFDE242	C/E/S/T/F	512 sequential file reads using an operand that is a scalar integer variable. Input from the file consists of 512 integers per iteration.
AFDE243	C/E/S/T/F	1 sequential file read using an operand that is a 512 element array with floating point components. Input from the file consists of 512 floating point numbers per iteration.
AFDE244	C/E/S/T/F	512 sequential file reads using an operand that is a floating point scalar variable. Input from the file consists of 512 floating point numbers per iteration.
AFDE245	C/E/S/T/F	1 sequential file read using an operand that is a 512 element array with character components. Input from the file consists of 512 characters per iteration.
AFDE246	C/E/S/T/F	512 sequential file reads using an operand that is a scalar character variable. Input from the file consists of 512 characters per iteration.
AFDE401	C/E/S/T/F	1968 direct file writes using an operand that is a scalar character variable. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE402	C/E/S/T/F	1968 direct file writes using an operand that is a 1 element character array. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE403	C/E/S/T/F	240 direct file writes using an operand that is an 8 element array with character components. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE404	C/E/S/T/F	24 direct file writes using an operand that is an 82 element array of characters with embedded carriage control. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE405	C/E/S/T/F	1 direct file write using an operand that is a 1968 element array with character components. Output to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE406	C/E/S/T/F	8040 direct file writes using an operand that is a

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AFDE407	C/E/S/T/F	scalar character variable. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132). 8040 direct file writes using an operand that is a 1 element character array. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE408	C/E/S/T/F	960 direct file writes using an operand that is an 8 element array with character components. Output to the file is 7800 characters per iteration (including carriage control) which fills a standard page of computer paper (by default 60 x 132).
AFDE409	C/E/S/T/F	60 direct file writes using an operand that is an 82 element array of characters with embedded carriage control. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE40A	C/E/S/T/F	1 direct file write using an operand that is a 8040 element array with character components. Output to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE411	C/E/S/T/F	1 direct file write using an operand that is a 512 element array with integer components. The output to the file consists of 512 integers per iteration.
AFDE412	C/E/S/T/F	512 direct file writes using an operand that is a scalar integer variable. Output to the file consists of 512 integers per iteration.
AFDE413	C/E/S/T/F	1 direct file write using an operand that is a 512 element array with floating point components. The output to the file consists of 512 floating point numbers per iteration.
AFDE414	C/E/S/T/F	512 direct file writes using an operand that is a floating point scalar variable. Output to the file consists of 512 floating point numbers per iteration.
AFDE415	C/E/S/T/F	1 direct file write using an operand that is a 512 element array with character components. The output to the file consists of 512 characters per iteration.
AFDE416	C/E/S/T/F	512 direct file writes using an operand that is a scalar character variable. Output to the file consists of 512 characters per iteration.
AFDE431	C/E/S/T/F	1968 direct file reads using an operand that is a scalar character variable. Input to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE432	C/E/S/T/F	1968 direct file reads using an operand that is a 1 element character array. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE433	C/E/S/T/F	240 direct file reads using an operand that is an 8 element array with character components. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE434	C/E/S/T/F	24 direct file reads using an operand that is an 82 element array of characters with embedded carriage control. Input to the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).
AFDE435	C/E/S/T/F	1 direct file read using an operand that is a 1968 element array with character components. Input from the file is 1968 characters (including carriage control) per iteration which fills a DEC VT100 terminal screen (by default 24 x 80).

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
AFDE436	C/E/S/T/F	8040 direct file reads using an operand that is a scalar character variable. Input to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE437	C/E/S/T/F	8040 direct file reads using an operand that is a 1 element character array. Input from the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE438	C/E/S/T/F	960 direct file reads using an operand that is an 8 element array with character components. Input from the file is 7800 characters per iteration (including carriage control) which fills a standard page of computer paper (by default 60 x 132).
AFDE439	C/E/S/T/F	60 direct file reads using an operand that is an 82 element array of characters with embedded carriage control. Input to the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE43A	C/E/S/T/F	1 direct file read using an operand that is a 8040 element array with character components. Input from the file is 8040 characters per iteration which fills a standard page of computer paper (by default 60 x 132).
AFDE441	C/E/S/T/F	1 direct file read using an operand that is a 512 element array with integer components. Input from the file is a block of integers per iteration.
AFDE442	C/E/S/T/F	512 direct file reads using an operand that is a scalar integer variable. Input from the file consists of 512 integers per iteration.
AFDE443	C/E/S/T/F	1 direct file read using an operand that is a 512 element array with floating point components. Input from the file consists of 512 floating point numbers per iteration.
AFDE444	C/E/S/T/F	512 direct file reads using an operand that is a floating point scalar variable. Input from the file consists of 512 floating point numbers per iteration.
AFDE445	C/E/S/T/F	1 direct file read using an operand that is a 512 element array with character components. Input from the file consists of 512 characters per iteration.
AFDE446	C/E/S/T/F	512 direct file reads using an operand that is a scalar character variable. Input from the file consists of 512 characters per iteration.
AFDF000	C/E/S/T/J/F	2 assignment statements of the form $x := \sin(y)$ where x, y are scalar variables.
AFDF001	C/E/S/T/J/F	2 assignment statements of the form $x := \operatorname{atan}(y)$ where x, y are scalar variables.
AFDF002	C/E/S/T/J/F	2 assignment statements of the form $x := \cos(y)$ where x, y are scalar variables.
AFDF003	C/E/S/T/J/F	2 assignment statements of the form $x := \operatorname{sqrt}(y)$ where x, y are scalar variables.
AFDF004	C/E/S/T/J/F	2 assignment statements of the form $x := \exp(y)$ where x, y are scalar variables.
AFDF005	C/E/S/T/J/F	2 assignment statements of the form $x := \ln(y)$ where x, y are scalar variables.
AGD0001	C/E/S/T/J/F	the Whetstone test as defined in the paper A Synthetic Benchmark published in the Computer Journal, Feb 1976 by Curnow and Michmann. Input/Output statements are not executed.
AGD0003	C/E/S/T	the Whetstone test as defined in the paper A Synthetic Benchmark published in the Computer Journal, Feb 1976 by Curnow and Michmann. The test uses predefined types integer and float for numeric types (compare to AGD0001 which uses <code>our_integer</code> and <code>our_float</code>). Input/Output statements are not executed.
AGD0009	C/E/S/T/J	the Dhrystone test as defined in the paper DHRYSTONE: A Synthetic Benchmark published in the CACM, Oct 1984 by

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
AG0000C	C/E/S/T/J	R.Micker. Run statistic #1 = size in bits of objects of type Record_Type(default discriminant).
AGD000E	C/E/S/T/J/F	ackermann's function as modified from the AdaTec Ada Fair test suite.
AGN000K	C/E/S/T/J	the Whetstone test as defined in the paper A Synthetic Benchmark published in the Computer Journal, Feb 1976 by Curnow and Michmann. Input/Output statements are not executed. external procedures are used rather than local procedures. Compare to AGD0001.
AGD0039,I	C/E/S/T	the Dhrystone test as defined in the paper DHRYSTONE: A Synthetic Benchmark published in the CACM, Oct 1984 by R.Micker. The test makes multiple microsecond level measurements of the Dhrystone test without initialization.
AGD0049,I	C/E/S/T	execution of the Dhrystone test by 10 subordinate tasks. Run statistic #1 = storage_size attribute of the subordinate task type. Compare to AGD0009 to see the effect of multi-tasking. Run statistics 1-4 in AGI0039 contain the total, maximum, minimum, median task iteration counts.
AGD0059,I	C/E/S/T	execution of the Dhrystone test by 100 subordinate tasks. Run statistic #1 = storage_size attribute of the subordinate task type. Compare to AGD0009 to see the effect of multi-tasking. Run statistics 1-4 in AGI0049 contain the total, maximum, minimum, median task iteration counts.
AL09101	C/E/S/T	execution of the Dhrystone test by 100 subordinate tasks with forced time-slicing(VAX=.01 secs.). Compare to AGD0049 to see the effect of forced time-slicing. Compare to AGD0009 to see the effect of multi-tasking. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 in AGI0049 contain the total, maximum, minimum, median task iteration counts.
AL09111	C/E/S/T	execution of a workload to randomly access a 128 element integer array with no subordinate tasks and without a priority pragma.
AL09121	C/E/S/T	execution of a workload to randomly access a 12800 element integer array with no subordinate tasks and without a priority pragma.
AL09131	C/E/S/T	execution of a workload to randomly access a 64000 element integer array with no subordinate tasks and without a priority pragma. Compare to test AL09101.
ALD9202	C/E/S/T	execution of a workload to randomly access a 128000 element integer array with no subordinate tasks and without a priority pragma. Compare to test AL09101.
ALD9203,I	C/E/S/T	execution of a workload to randomly access a 128 element integer array by 1 subordinate task. Compare to test AL09101 to see the effect of task scheduling. Run statistic #1 = storage_size attribute of the subordinate task type.
ALD9204,I	C/E/S/T	execution of a workload to randomly access a 128 element integer array by 10 subordinate tasks. Compare to AL09101,ALD9202 to see the effect of multi-tasking. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9203 contain the total, maximum, minimum, median task iteration counts.
		execution of a workload to randomly access a 128 element integer array by 10 subordinate tasks with forced time-slicing. Compare to ALD9203 to see the effect of forced time-slicing. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9204 contain the total, maximum, minimum, median task iteration counts.

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
ALD9212	C/E/S/T	execution of a workload to randomly access a 12800 element integer array by 1 subordinate task. Compare to test AL09111 to see the effect of task scheduling. Run statistic #1 = storage_size attribute of the subordinate task type.
ALD9213,I	C/E/S/T	execution of a workload to randomly access a 12800 element integer array by 10 subordinate tasks. Compare to AL09111,ALD9212 to see the effect of multi-tasking. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9213 contain the total,maximum,minimum,median task iteration counts.
ALD9214,I	C/E/S/T	execution of a workload to randomly access a 12800 element integer array by 10 subordinate tasks with forced time-slicing. Compare to ALD9213 to see the effect of forced time-slicing. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9214 contain the total,maximum,minimum,median task iteration counts.
ALD9222	C/E/S/T	execution of a workload to randomly access a 64000 element integer array by 1 subordinate task. Compare to test AL09121 to see the effect of task scheduling. Run statistic #1 = storage_size attribute of the subordinate task type.
ALD9223,I	C/E/S/T	execution of a workload to randomly access a 64000 element integer array by 10 subordinate tasks. Compare to AL09121,ALD9222 to see the effect of multi-tasking. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9223 contain the total,maximum,minimum,median task iteration counts.
ALD9224,I	C/E/S/T	execution of a workload to randomly access a 64000 element integer array by 10 subordinate tasks with forced time-slicing. Compare to ALD9223 to see the effect of forced time-slicing. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9224 contain the total,maximum,minimum,median task iteration counts.
ALD9232	C/E/S/T	execution of a workload to randomly access a 128000 element integer array by 1 subordinate task. Compare to test AL09131 to see the effect of task scheduling. Run statistic #1 = storage_size attribute of the subordinate task type.
ALD9233,I	C/E/S/T	execution of a workload to randomly access a 128000 element integer array by 10 subordinate tasks. Compare to AL09131,ALD9232 to see the effect of multi-tasking. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9233 contain the total,maximum,minimum,median task iteration counts.
ALD9234,I	C/E/S/T	execution of a workload to randomly access a 128000 element integer array by 10 subordinate tasks with forced time-slicing. Compare to ALD9233 to see the effect of forced time-slicing. Run statistic #1 = storage_size attribute of the subordinate task type. Run statistics 1-4 of ALI9234 contain the total,maximum,minimum,median task iteration counts.
ALDE2C1	C/E/S/T	execution of a work-load by 1 subordinate task. The work load outputs 1968 characters to a sequential file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C2,I	C/E/S/T	execution of a work-load by 10 subordinate tasks. Compare to ALDE2C1 to see the effect of task scheduling overhead. The work load outputs 1968 characters to a sequential file of character type. Run statistics 1-4

Table F-1 ACPs Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
ALDE2C3	C/E/S/T	in ALIE2C2 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type). execution of a work-load by 1 subordinate task with a forced timeslice interval (VAX=.01 secs). Compare to ALDE2C1 to see the effect of forced time-slicing. The work load outputs 1968 characters to a sequential file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C4,I	C/E/S/T	execution of a work-load by 10 subordinate tasks with a forced time-slice interval (VAX .01 secs). Compare to ALDE2C2 to see the effect of forced time-slicing. The work load outputs 1968 characters to a sequential file of character type. Run statistics 1-4 in ALIE2C4 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C5	C/E/S/T	execution of a work-load by 1 subordinate task. The work load inputs 1968 characters from a sequential file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C6,I	C/E/S/T	execution of a work-load by 10 subordinate tasks. Compare to ALDE2C5 to see the effect of task scheduling overhead. The work load inputs 1968 characters from a sequential file of character type. Run statistics 1-4 in ALIE2C6 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C7	C/E/S/T	execution of a work-load by 1 subordinate task with a forced timeslice interval (VAX=.01 secs). Compare to ALDE2C5 to see the effect of forced time-slicing. The work load inputs 1968 characters from a sequential file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE2C8,I	C/E/S/T	execution of a work-load by 10 subordinate tasks with a forced time-slice interval (VAX .01 secs). Compare to ALDE2C6 to see the effect of forced time-slicing. The work load inputs 1968 characters from a sequential file of character type. Run statistics 1-4 in ALIE2C8 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C1	C/E/S/T	execution of a work-load by 1 subordinate task. to ALDE4C1 to see the effect of task scheduling overhead. The work load outputs 1968 characters to a direct file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C2,I	C/E/S/T	execution of a work-load by 10 subordinate tasks. Compare to ALDE4C1 to see the effect of task scheduling overhead. The work load outputs 1968 characters to a direct file of character type. Run statistics 1-4 in ALIE4C2 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C3	C/E/S/T	execution of a work-load by 1 subordinate task with a forced timeslice interval (VAX=.01 secs). Compare to ALDE4C1 to see the effect of forced time-slicing. The work load outputs 1968 characters to a direct file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C4,I	C/E/S/T	execution of a work-load by 10 subordinate tasks with a forced time-slice interval (VAX .01 secs). Compare to ALDE4C2 to see the effect of forced time-slicing. The work load outputs 1968 characters to a direct file of character type. Run statistics 1-4 in ALIE4C4 contain

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
ALDE4C5	C/E/S/T	the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type). execution of a work-load by 1 subordinate task. The work load inputs 1968 characters from a direct file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C6,I	C/E/S/T	execution of a work-load by 10 subordinate tasks. Compare to ALDE4C5 to see the effect of task scheduling overhead. The work load inputs 1968 characters from a direct file of character type. Run statistics 1-4 in ALDE4C6 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C7	C/E/S/T	execution of a work-load by 1 subordinate task with a forced timeslice interval (VAX=.01 secs). Compare to ALDE4C5 to see the effect of forced time-slicing. The work load inputs 1968 characters from a direct file of character type. (run statistic #1 = storage_size attribute of the subordinate task type).
ALDE4C8,I	C/E/S/T	execution of a work-load by 10 subordinate tasks with a forced time-slice interval (VAX .01 secs). Compare to ALDE4C6 to see the effect of forced time-slicing. The work load inputs 1968 characters from a direct file of character type. Run statistics 1-4 in ALDE4C8 contain the total, maximum, minimum, and median task iteration counts. (run statistic #1 = storage_size attribute of the subordinate task type).
A000102	C/E/S/T/J/F	hand-optimized version of test A000103.
A000103,C	C/E/S/T/J/F	tests constant folding optimization logic for constant scalar and character expressions within assignment statements.
A000107	C/E/S/T	hand-optimized version of test A000108.
A000108,C	C/E/S/T	tests constant folding/propagation optimization logic for constant aggregate expressions within/between assignment statements.
A000112	C/E/S/T	hand-optimized version of test A000113.
A000113,C	C/E/S/T	tests constant folding optimization logic for constant predefined attribute expressions within assignment statements.
A000202	C/E/S/T/J/F	hand-optimized version of test A000203.
A000203,C	C/E/S/T/J/F	tests constant propagation optimization logic for scalar and character constant expressions within and between assignment and control structure statements by providing opportunities for removal of statements at compile-time.
A000207	C/E/S/T	hand-optimized version of test A000208.
A000208,C	C/E/S/T	tests constant propagation optimization logic for aggregate constant expressions within and between assignment and control structure statements by providing opportunities for removal of statements at compile-time.
A000300	C/E/S/T/J/F	tests common array address expression optimization logic for 1-dimensional integer arrays.
A000305	C/E/S/T/J/F	tests common array address expression optimization logic for 2 dimensional integer arrays.
A000310	C/E/S/T/J/F	tests common array address expression optimization logic for 3 dimensional integer arrays.
A000311	C/E/S/T/J/F	hand-optimized version of test A000312.
A000312,C	C/E/S/T/J/F	tests common subexpression optimization logic within/between assignment statements for arithmetic expressions with integer scalar operands.
A000314	C/E/S/T/J/F	hand-optimized version of test A000315.
A000315,C	C/E/S/T/J/F	tests common subexpression optimization logic within/between assignment statements for arithmetic

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
A000317 A000318,C	C/E/S/T/F C/E/S/T/F	expressions with 1-dimensional integer array element operands. hand-optimized version of test A000318. tests common subexpression optimization logic within/between assignment statements for arithmetic expressions with 2-dimensional integer array element operands.
A000320 A000321,C	C/E/S/T/F C/E/S/T/F	hand-optimized version of test A000321. tests common subexpression optimization logic within/between assignment statements for arithmetic expressions with 3-dimensional integer array element operands.
A000323 A000324,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000324. tests common subexpression optimization logic within and between assignment and control statements for arithmetic expressions with integer scalar operands.
A000326 A000327,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000327. tests common subexpression optimization logic across short-circuit operators in IF statements for arithmetic expressions involving integer scalar operands.
A000329 A000330,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000330. tests common subexpression optimization logic for common 1-dimensional array element references within a FOR loop with the loop iteration parameter used as index.
A000332 A000333,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000333. tests common subexpression optimization logic for common 2-dimensional array element references within a 2-level nested FOR loop with the loop iteration parameters used as indices.
A000335 A000336,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000336. tests common subexpression optimization logic for common 3-dimensional array element references within a 3-level nested FOR loop with the loop iteration parameters used as array indices.
A000338 A000339,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000339. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with integer scalar operands.
A000341 A000342,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000342. tests common subexpression optimization logic within/between assignment statements for expressions using the mod,rem and abs operators.
A000344 A000345,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000345. tests common subexpression optimization logic within/between assignment statements for expressions using the math functions sin,cos,tan.
A000348 A000349,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000349. tests common subexpression optimization logic between assignment statements and across a call to an external procedure. The expressions involve local variables that are not referenced by nor passed as arguments to the called procedure.
A000351 A000352,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000352. tests common subexpression optimization logic between assignment statements and across a call to an external procedure. The expressions involve global variables that are not referenced by nor passed as arguments to the called procedure.
A000354 A000355,C	C/E/S/T/F C/E/S/T/F	the test requires range checks for all indices of array element references. It is compared to test A000355. tests array index range check optimization logic by assigning constants (within separate paths of an IF statement) to variables used in subsequent array references. No array index range checks are necessary.

Table F-1 ACPS Test Descriptions (continued)

<u>TEST NAME</u>	<u>VERSIONS</u>	<u>DESCRIPTION</u>
A000357	C/E/S/T/F	If optimized, A000355 should be < 1. the test requires range checks for all indices of array element references. It is compared to test A000358. tests array index range check optimization logic by assigning positive valued variables (within an IF statement) to variables used in subsequent array references. Only lower bound array index range checks are necessary. If optimized, A000358 should be < 1.
A000358,C	C/E/S/T/F	
A000360	C/E/S/T/F	the test requires range checks for all indices of array element references. It is compared to test A000361. tests array index range check optimization logic by assigning (from within an IF statement) variables typed with ranges to variables typed without ranges which are used in subsequent array references. No array index range checks are necessary. If optimized, A000361 should be < 1.
A000361,C	C/E/S/T/F	
A000363	C/E/S/T/J	hand-optimized version of test A000364. Tests A000364, A000367, A00036A, A00036D, A00036G, A00036J, A00036M, A00036P, A00036S, A00036V, A00036Y all have the same structure and may be compared to one another to see whether the type of the operand used in the tests affects optimization.
A000364,C	C/E/S/T/J	tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with enumeration scalar operands.
A000366	C/E/S/T	hand-optimized version of test A000367. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with integer - array slice operands.
A000367,C	C/E/S/T	
A000369	C/E/S/T	hand-optimized version of test A00036A. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with boolean - array slice operands.
A00036A,C	C/E/S/T	
A00036C	C/E/S/T	hand-optimized version of test A00036D. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with packed boolean - array slice operands.
A00036D,C	C/E/S/T	
A00036F	C/E/S/T	hand-optimized version of test A00036G. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with record object operands (type RECORDS).
A00036G,C	C/E/S/T	
A00036I	C/E/S/T/J	hand-optimized version of test A00036J. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with operands that are access type components of record access objects (type RECORD_POINTER).
A00036J,C	C/E/S/T/J	
A00036L	C/E/S/T/J	hand-optimized version of test A00036M. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with operands that are integer components of record objects (type RECORDS).
A00036M,C	C/E/S/T/J	
A00036O	C/E/S/T/J	hand-optimized version of test A00036P. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with operands that are integer components of record objects (type PACKED_RECORDS).
A00036P,C	C/E/S/T/J	
A00036R	C/E/S/T/J	hand-optimized version of test A00036S. tests common subexpression optimization logic within and
A00036S,C	C/E/S/T/J	

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
A00036U A00036V,C	C/E/S/T/J C/E/S/T/J	between assignment and control statements for relational and logical operators with operands that are integer components of record objects (type REP_RECORDS). hand-optimized version of test A00036V. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with operands that are string components of record access objects (type RECORD_TYPE).
A00036X A00036Y,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A00036Y. tests common subexpression optimization logic within and between assignment and control statements for relational and logical operators with short_integer scalar operands.
A000370 A000371,C	C/E/S/T C/E/S/T	non-shared global variable version of test A000371. tests common subexpression optimization logic within/between assignment statements for arithmetic expressions with integer scalar operands. The test is executed within an Ada task and the variables in the expressions are declared as shared variables. A000371 compares optimization logic for non-shared variables to that for shared variables.
A000375 A000376,C	C/E/S/T C/E/S/T	hand-optimized version of test A000376. tests common subexpression optimization logic for arithmetic expressions with integer scalar operands across a call to a task entry. The variables in the expressions are not referenced by the rendezvous. The test is executed within an Ada task and the variables in the expressions are declared as global variables.
A000380 A000381,C	C/E/S/T C/E/S/T	hand-optimized version of test A000381. tests common subexpression optimization logic for arithmetic expressions with integer scalar operands across a call to a task entry. The variables in the expressions are not referenced by the rendezvous. The test is executed within an Ada task and the variables in the expressions are declared as shared variables. Compare A000381 to A000376.
A000391	C/E/S/T/J/F	tests optimization logic to recognize common argument lists in external procedure calls and to eliminate any redundant code/data. The test contains 5 procedure calls with identical argument lists consisting of 10 1-dimensional integer array input/output arguments. The extent of optimization is best determined through examination of the compiled code.
A000402 A000403,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000403. tests code motion space optimization logic to conserve space by moving code common to all paths in IF statements.
A000500	C/E/S/T/J/F	tests dead assignment optimization logic by providing opportunities to eliminate assignment statements to local variables whose values are only used within the assignment statements. All statements in the test loop can be removed.
A000502 A000503,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000503. tests dead assignment optimization logic by providing opportunities to eliminate assignment statements to global variables that are not used but always set in a local procedure called within the test loop.
A000507 A000508,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000508. tests dead assignment optimization logic by providing opportunities to eliminate assignment statements to global variables that are not used but always set as output parameters to a local procedure called within the test loop.
A000512	C/E/S/T/J/F	hand-optimized version of test A000513.

Table F-1 ACPS Test Descriptions (continued)

TEST NAME	VERSIONS	DESCRIPTION
A000513,C	C/E/S/T/J/F	tests dead assignment optimization logic by providing opportunities to eliminate redundant assignment statements within a sequential set of assignment and control structure statements.
A000517 A000518,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000518. tests dead assignment optimization logic by providing opportunities to eliminate redundant assignment statements within a sequential set of assignment statements.
A0D0523 A0D0524,C	C/E/S/T/J C/E/S/T/J	hand-optimized version of test A0D0524. tests dead assignment optimization logic by providing opportunities to eliminate assignment statements to global variables that are not used but always set in a local pragma inline procedure called within the test loop.
A0D0527 A0D0528,C	C/E/S/T C/E/S/T	hand-optimized version of test A0D0528. tests dead assignment optimization logic by providing opportunities to eliminate assignment statements to global variables that are not used but always set in an external pragma inline procedure called within the test loop.
A000602 A000603,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000603. tests algebraic identity recognition optimization logic by providing opportunities to simplify arithmetic and relational expressions within assignment statements.
A000605	C/E/S/T/J/F	tests target dependent operator strength reduction optimization logic for integer expressions within assignment statements(see test code for run statistic description).
A000606	C/E/S/T/J/F	tests target dependent operator strength reduction logic for multiplication/division by powers of 2(e.g. 4,8,...).
A000607	C/E/S/T/J/F	tests target dependent operator strength reduction logic for exponentiation by constant exponents(e.g. 2,3,4,8).
A000702 A000703,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000703. tests optimization logic to remove loop invariant code(expression,assignment statement,control statement) from for loops.
A000705	C/E/S/T/J/F	tests optimization logic to remove loop invariant array index calculations from FOR loops.
A000706 A000707,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000707. tests optimization logic to remove loop invariant expressions involving array references from FOR loops.
A000709 A000710,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000710. tests loop fusion optimization logic by providing an opportunity to combine 2 loops into 1.
A000713	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that transfer data between 1-dimensional arrays.
A000714	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that transfer data between 2-dimensional arrays.
A000715	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that transfer data between 3-dimensional arrays.
A000716	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that initialize 1-dimensional arrays to a constant.
A000717	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that initialize 2-dimensional arrays to a constant.
A000718	C/E/S/T/J/F	tests target machine dependent optimization logic to use block move instructions to implement FOR loops that initialize 3-dimensional arrays to a constant.
A000719 A000720,C	C/E/S/T/J/F C/E/S/T/J/F	hand-optimized version of test A000720. tests time optimization logic to unroll a 3-iteration FOR loop.

APPENDIX G
TEST PROGRAMS AND SOURCE CODE FILES

Table G-1 Ada Type A Test Programs and Source Code Files

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
AA00000	OURSYS*	OURDMP*	AA0P000			
AF03519	OURSYS*	OURDMP*	AF03500	AF03501	AF03502	AF03503
	AF03504	AF03509	AF03510	AF03511	AF03512	AF03513
	AF03514	AF03517	AF03518	AF0P000		
AF03550	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD3551	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF03620	OURSYS*	OURDMP*	AF03600	AF03601	AF03602	AF03603
	AF03604	AF03605	AF03606	AF03607	AF03608	AF03609
	AF03610	AF03611	AF03612	AF03613	AF03614	AF03615
	AF03616	AF03617	AF03618	AF03619	AF0P000	
AF03650	OURSYS*	OURDMP*	AF03630	AF03631	AF03632	AF03633
	AF03634	AF03635	AF03636	AF03637	AF03638	AF03639
	AF03641	AF03642	AF03643	AF03644	AF03645	AF03646
	AF03647	AF03648	AF03649	AF0P000		
AF03704	OURSPC*	OURSYS*	OURDMP*	AF03700	AF03701	AF03702
	AF03703	AF0P000				
AF03805	OURSPC*	OURSYS*	OURDMP*	AF03800	AF03801	AF03802
	AF03803	AF03804	AF0P000			
AF04120	OURSYS*	OURDMP*	AF0P000			
AF04121	OURSYS*	OURDMP*	AF0P000			
AF04122	OURSYS*	OURDMP*	AF0P000			
AF04123	OURSYS*	OURDMP*	AF0P000			
AF04124	OURSYS*	OURDMP*	AF0P000			
AF04125	OURSYS*	OURDMP*	AF0P000			
AF04126	OURSYS*	OURDMP*	AF0P000			
AF04127	OURSYS*	OURDMP*	AF0P000			
AFD4128*	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04129	OURSYS*	OURDMP*	AF0P000			
AFD412A	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0412B	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD412C	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD412D	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04130	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04131	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04132	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04133	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD4135	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD4136	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD4137	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD4138	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD4139	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD413A	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD413B	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD413C	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0413D	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD413E	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD413G	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0413H	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0413I	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0413J	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0413K	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04310	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04311	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04312	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF04510	OURSYS*	OURDMP*	AF0P000			
AF04511	OURSYS*	OURDMP*	AF0P000			
AF04512	OURSYS*	OURDMP*	AF0P000			
AF04513	OURSYS*	OURDMP*	AF0P000			
AF04514	OURSYS*	OURDMP*	AF0P000			
AF04515	OURSYS*	OURDMP*	AF0P000			
AF04516	OURSYS*	OURDMP*	AF0P000			
AF04517	OURSYS*	OURDMP*	AF0P000			
AF0451J	OURSYS*	OURDMP*	AF04518	AF04519	AF0451A	AF0451B
	AF0451C	AF0451D	AF0451E	AF0451F	AF0451G	AF0451H
	AF0451I	AF0P000				

* Source file supplied with USE file type only

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
AFD451U*	OURSYS*	OURDMP*	OURTYP*	AFOP000	AFD451K	AFD451L
	AFD451M	AFD451N	AFD451O	AFD451P	AFD451Q	AFD451R
	AFD451S					
AF0451Z	OURSYS*	OURDMP*	AF0451V	AF0451W	AF0451X	AF0451Y
	AFOP000					
AF04520	OURSYS*	OURDMP*	AFOP000			
AF04521	OURSYS*	OURDMP*	AFOP000			
AF04522	OURSYS*	OURDMP*	AFOP000			
AFD4523	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4524	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4525	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4526	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF04527	OURSYS*	OURDMP*	AFOP000			
AFD4528	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF04529	OURSYS*	OURDMP*	AFOP000			
AFD452A	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF04530	OURSYS*	OURDMP*	AFOP000			
AF04531	OURSYS*	OURDMP*	AFOP000			
AF04532	OURSYS*	OURDMP*	AFOP000			
AF04533	OURSYS*	OURDMP*	AFOP000			
AF04534	OURSYS*	OURDMP*	AFOP000			
AF04535	OURSYS*	OURDMP*	AFOP000			
AF04536	OURSYS*	OURDMP*	AFOP000			
AF04537	OURSYS*	OURDMP*	AFOP000			
AF04538	OURSYS*	OURDMP*	AFOP000			
AF04539	OURSYS*	OURDMP*	AFOP000			
AF0453A	OURSYS*	OURDMP*	AFOP000			
AF0453B	OURSYS*	OURDMP*	AFOP000			
AF0453C	OURSYS*	OURDMP*	AFOP000			
AF04540	OURSYS*	OURDMP*	AFOP000			
AF04541	OURSYS*	OURDMP*	AFOP000			
AF04550	OURSYS*	OURDMP*	AFOP000			
AF04551	OURSYS*	OURDMP*	AFOP000			
AF04552	OURSYS*	OURDMP*	AFOP000			
AF04553	OURSYS*	OURDMP*	AFOP000			
AF04554	OURSYS*	OURDMP*	AFOP000			
AF04555	OURSYS*	OURDMP*	AFOP000			
AF04556	OURSYS*	OURDMP*	AFOP000			
AF04557	OURSYS*	OURDMP*	AFOP000			
AF04558	OURSYS*	OURDMP*	AFOP000			
AF04559	OURSYS*	OURDMP*	AFOP000			
AFD455D	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455E	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455F	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455G	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455I	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455J	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455K	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD455L	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF0455N	OURSYS*	OURDMP*	AFOP000			
AF0455O	OURSYS*	OURDMP*	AFOP000			
AF04560	OURSYS*	OURDMP*	AFOP000			
AF04562	OURSYS*	OURDMP*	AFOP000			
AF04563	OURSYS*	OURDMP*	AFOP000			
AFD4566	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4567	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4568	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4569	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF04600	OURSYS*	OURDMP*	AFOP000			
AF04601	OURSYS*	OURDMP*	AFOP000			
AF04602	OURSYS*	OURDMP*	AFOP000			
AFD4603	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4604	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4605	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4606	OURSYS*	OURDMP*	OURTYP*	AFOP000		

* Source file supplied with USE file type only

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
AFD4607	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4608	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD4609	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD460A	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD460B	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF0460E	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF0460F	OURSYS*	OURDMP*	AFOP000			
AF0460G	OURSYS*	OURDMP*	AFOP000			
AF0460H	OURSYS*	OURDMP*	AFOP000			
AF0460I	OURSYS*	OURDMP*	AFOP000			
AF0460J	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD460K	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD460L	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF0460M	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD460N	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD460O	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD4804	OURSPC*	OURSYS*	OURDMP*	AFOP000	AFD4800	AFD4801
	AFD4802	AFD4803**				
AF05200	OURSYS*	OURDMP*	AFOP000			
AF05201	OURSYS*	OURDMP*	AFOP000			
AF05202	OURSYS*	OURDMP*	AFOP000			
AF05203	OURSYS*	OURDMP*	AFOP000			
AF05204	OURSYS*	OURDMP*	AFOP000			
AF05205	OURSYS*	OURDMP*	AFOP000			
AF05206	OURSYS*	OURDMP*	AFOP000			
AF05207	OURSYS*	OURDMP*	AFOP000			
AF05208	OURSYS*	OURDMP*	AFOP000			
AF05209	OURSYS*	OURDMP*	AFOP000			
AF0520A	OURSYS*	OURDMP*	AFOP000			
AF0520B	OURSYS*	OURDMP*	AFOP000			
AFD520C	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520D	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520E	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520F	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520G	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520H	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520I	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD520J	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF0520M	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD520N	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF0520O	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD520P	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AFD520Q	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF0520R	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF0520S	OURSPC*	OURSYS*	OURDMP*	AFOP000		
AF05210	OURSYS*	OURDMP*	AFOP000			
AF05211	OURSYS*	OURDMP*	AFOP000			
AF05212	OURSYS*	OURDMP*	AFOP000			
AF05213	OURSYS*	OURDMP*	AFOP000			
AF05214	OURSYS*	OURDMP*	AFOP000			
AF05215	OURSYS*	OURDMP*	AFOP000			
AFD5218	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD5219	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521A	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521B	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521C	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521D	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521E	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AFD521F	OURSYS*	OURDMP*	OURTYP*	AFOP000		
AF0521H	OURSYS*	OURDMP*	AFOP000			
AF0521I	OURSYS*	OURDMP*	AFOP000			
AF0521J	OURSYS*	OURDMP*	AFOP000			
AF0521K	OURSYS*	OURDMP*	AFOP000			
AFD521K	OURSPC*	OURSYS*	OURDMP*	AFOP000		

** Multiple versions of source file supplied(USE and ADA file types)

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
AF0521L	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD521M	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF0521N	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD521O	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AFD521P	OURSPC*	OURSYS*	OURDMP*	AF0P000		
AF05304	OURSYS*	OURDMP*	AF05300	AF05301	AF05302	AF05303
	AF0P000					
AF05308	OURSYS*	OURDMP*	AF05305	AF05306	AF05307	AF0P000
AF05408	OURSYS*	OURDMP*	AF05400	AF05401	AF05402	AF05403
	AF05404	AF05405	AF05406	AF05407	AF0P000	
AF05505	OURSYS*	OURDMP*	AF05501	AF05502	AF05503	AF05504
	AF0P000					
AF0550C	OURSYS*	OURDMP*	AF05506	AF05507	AF05508	AF05509
	AF0550A	AF0550B	AF0P000			
AF0550D	OURSYS*	OURDMP*	AF0P000			
AF0550E	OURSYS*	OURDMP*	AF0P000			
AF0550F	OURSYS*	OURDMP*	AF0P000			
AF06001	OURSYS*	OURDMP*	AF0P000			
AF06009	OURSYS*	OURDMP*	AF0P000			
AF06010	OURSYS*	OURDMP*	AF0P000			
AF06011	OURSYS*	OURDMP*	AF0P000			
AF06022	OURSYS*	OURDMP*	AF06013	AF06014	AF06015	AF06016
	AF06017	AF06018	AF06019	AF0P000		
AF06033	OURDMP*	AF06023	AF06024	AF06025	AF06026	
	AF06027	AF06028	AF06029	AF06030	AF06031	AF06032
	AF0P000					
AF06053	OURSYS*	OURDMP*	AF06043	AF06044	AF06045	AF06046
	AF06047	AF06048	AF06049	AF06050	AF06051	AF06052
	AF0P000					
AF06069	OURSYS*	OURDMP*	AF06060	AF06061	AF06062	AF06063
	AF06064	AF06065	AF06066	AF06067	AF06068	AF0P000
AF06079	OURSYS*	OURDMP*	AF06070	AF06071	AF06072	AF06073
	AF06074	AF06075	AF06076	AF06077	AF06078	AF0P000
AF06101	OURSYS*	OURDMP*	AF06100	AF0P000		
AF06109	OURSYS*	OURDMP*	AF06108	AF0P000		
AF06110	OURSYS*	OURDMP*	AF06108	AF0P000		
AF06111	OURSYS*	OURDMP*	AF06108	AF0P000		
AF06122	OURSYS*	OURDMP*	AF06112	AF06113	AF06114	AF06115
	AF06116	AF06117	AF06118	AF06119	AF0P000	
AF06140	OURSYS*	OURDMP*	AF06132	AF06133	AF06134	AF06135
	AF06136	AF06137	AF06138	AF06139	AF0P000	
AF06150	OURSYS*	OURDMP*	AF06142	AF06143	AF06144	AF06145
	AF06146	AF06147	AF06148	AF06149	AF0P000	
AF06160	OURSYS*	OURDMP*	AF06152	AF06153	AF06154	AF06155
	AF06156	AF06157	AF06158	AF06159	AF0P000	
AF06170	OURSYS*	OURDMP*	AF06162	AF06163	AF06164	AF06165
	AF06166	AF06167	AF06168	AF06169	AF0P000	
AF06180	OURSYS*	OURDMP*	AF06172	AF06173	AF06174	AF06175
	AF06176	AF06177	AF06178	AF06179	AF0P000	
AF06190	OURSYS*	OURDMP*	AF06182	AF06183	AF06184	AF06185
	AF06186	AF06187	AF06188	AF06189	AF0P000	
AF06199	OURSPC*	OURSYS*	OURDMP*	AF06191	AF06192	AF06193
	AF06194	AF06195	AF06196	AF06197	AF06198	AF0P000
AF0619I	OURSPC*	OURSYS*	OURDMP*	AF0619A	AF0619B	AF0619C
	AF0619D	AF0619E	AF0619F	AF0619G	AF0619H	AF0P000
AF0619R	OURSPC*	OURSYS*	OURDMP*	AF0619J	AF0619K	AF0619L
	AF0619M	AF0619N	AF0619O	AF0619P	AF0619Q	AF0P000
AFD6201	OURSYS*	OURDMP*	AF0P000			
AFD6209	OURSYS*	OURDMP*	AF0P000			
AFD6210	OURSYS*	OURDMP*	AF0P000			
AFD6211	OURSYS*	OURDMP*	AF0P000			
AFD6222	OURSYS*	OURDMP*	AF0P000	AFD6213	AFD6214	AFD6215
	AFD6216	AFD6217	AFD6218	AFD6219		
AFD6233	OURSYS*	OURDMP*	AF0P000	AFD6223	AFD6224	AFD6225
	AFD6226	AFD6227	AFD6228	AFD6229	AFD6230	AFD6231

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)					
AFD6253	AFD6232 OURSYSX AFD6246 AFD6252	OURDMPX AFD6247	AF0P000 AFD6248	AFD6243 AFD6249	AFD6244 AFD6250	AFD6245 AFD6251
AFD6269	OURSYSX AFD6263	OURDMPX AFD6264	AF0P000 AFD6265	AFD6260 AFD6266	AFD6261 AFD6267	AFD6262 AFD6268
AFD6279	OURSYSX AFD6273	OURDMPX AFD6274	AF0P000 AFD6275	AFD6270 AFD6276	AFD6271 AFD6277	AFD6272 AFD6278
AFD6301	OURSYSX	OURDMPX	AF0P000	AFD6300		
AFD6309	OURSYSX	OURDMPX	AF0P000	AFD6308		
AFD6310	OURSYSX	OURDMPX	AF0P000	AFD6308		
AFD6311	OURSYSX	OURDMPX	AF0P000	AFD6308		
AFD6322	OURSYSX AFD6315	OURDMPX AFD6316	AF0P000 AFD6317	AFD6312 AFD6318	AFD6313 AFD6319	AFD6314
AFD6340	OURSYSX AFD6335	OURDMPX AFD6336	AF0P000 AFD6337	AFD6332 AFD6338	AFD6333 AFD6339	AFD6334
AFD6350	OURSYSX AFD6345	OURDMPX AFD6346	AF0P000 AFD6347	AFD6342 AFD6348	AFD6343 AFD6349	AFD6344
AFD6360	OURSYSX AFD6355	OURDMPX AFD6356	AF0P000 AFD6357	AFD6352 AFD6358	AFD6353 AFD6359	AFD6354
AFD6370	OURSYSX AFD6365	OURDMPX AFD6366	AF0P000 AFD6367	AFD6362 AFD6368	AFD6363 AFD6369	AFD6364
AFD6380	OURSYSX AFD6375	OURDMPX AFD6376	AF0P000 AFD6377	AFD6372 AFD6378	AFD6373 AFD6379	AFD6374
AFD6390	OURSYSX AFD6385	OURDMPX AFD6386	AF0P000 AFD6387	AFD6382 AFD6388	AFD6383 AFD6389	AFD6384
AF0642B	OURSPCX AF06425 AF0P000	OURSYSX AF06426	OURDMPX AF06427	AF06422 AF06428	AF06423 AF06429	AF06424 AF0642A
AF06803	OURSYSX	OURDMPX	AF06802	AF0P000		
AF06807	OURSYSX	OURDMPX	AF06806	AF0P000		
AF06809	OURSPCX	OURSYSX	OURDMPX	AF06808	AF0P000	
AF0680B	OURSPCX	OURSYSX	OURDMPX	AF0680A	AF0P000	
AF0680D	OURSPCX	OURSYSX	OURDMPX	AF0680C	AF0P000	
AF0680F	OURSPCX	OURSYSX	OURDMPX	AF0680E	AF0P000	
AFD680H	OURSPCX	OURSYSX	OURDMPX	AF0P000	AFD680G	
AFD680J	OURSPCX	OURSYSX	OURDMPX	AF0P000	AFD680I	
AF06811	OURSYSX	OURDMPX	AF06810	AF0P000		
AF06815	OURSYSX	OURDMPX	AF06814	AF0P000		
AF06817	OURSYSX	OURDMPX	AF06816	AF0P000		
AF06819	OURSYSX	OURDMPX	AF06818	AF0P000		
AF06821	OURSYSX	OURDMPX	AF06820	AF0P000		
AF06823	OURSPCX	OURSYSX	OURDMPX	AF06822	AF0P000	
AF06825	OURSPCX	OURSYSX	OURDMPX	AF06824	AF0P000	
AFN9301	OURSPCX	OURSYSX	AFN9300			
AFN9302	OURSPCX	OURSYSX	AFN9300			
AFN9303	OURSPCX	OURSYSX	AFN9300			
AF09501	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09502	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09503	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09504	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09505	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09506	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09507	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09508	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AF09509	OURSPCX	OURSYSX	OURDMPX	AF09500	AF0P000	
AFN9511	OURSPCX	OURSYSX	AFN9510			
AF09600	OURSPCX	OURSYSX	AF0P000			
AF09601	OURSPCX	OURSYSX	AF0P000			
AF09602	OURSPCX	OURSYSX	AF0P000			
AF09603	OURSPCX	OURSYSX	AF0P000			
AF09604	OURSPCX	OURSYSX	AF0P000			
AF09605	OURSPCX	OURSYSX	AF0P000			
AF09606	OURSPCX	OURSYSX	AF0P000			
AF09607	OURSPCX	OURSYSX	AF0P000			

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
AFN9611	OURSPC*	OURSYS*				
AFN9612	OURSPC*	OURSYS*	<u>AFN9610</u>			
AFN9613	OURSPC*	OURSYS*	<u>AFN9610</u>			
AF09711	OURSYS*	AF09710	AF0P000			
AF09713	OURSYS*	AF09712	AF0P000			
AF09720	OURSPC*	OURSYS*	OURDMP*	AF09500	AF0P000	
AF09721	OURSPC*	OURSYS*	OURDMP*	AF09500	AF0P000	
AF09731	OURSPC*	OURSYS*	OURDMP*	AF09500	AF0P000	
AF09901	OURSYS*	AF09900	AF0P000			
AF09902	OURSYS*	AF09900	AF0P000			
AF09903	OURSYS*	AF09900	AF0P000			
AFM9A01	OURSPC*	OURSYS*	AFM9A00			
AFM9A02	OURSPC*	OURSYS*	AFM9A00			
AFM9A03	OURSPC*	OURSYS*	AFM9A00			
AF09B01	OURSPC*	OURSYS*	OURDMP*	AF09500	AF0P000	
AFD9C00	OURSPC*	OURSYS*	AF0P000	AFD9000		
AFD9C01	OURSPC*	OURSYS*	AF0P000	AFD9000	<u>AFD9200</u>	
<u>AFD9C02</u>	OURSPC*	OURSYS*	AF0P000	AFD9000	<u>AFD9200</u>	
<u>AFD9C03**</u>	OURSPC*	OURSYS*	AF0P000	AFD9000	<u>AFD9200</u>	
<u>AFD9C04**</u>	OURSPC*	OURSYS*	AF0P000	AFD9000	<u>AFD9200</u>	
AFMB001	OURSYS*	AFMB000				
AFMB003	OURSYS*	AFMB002				
AFMB005	OURSYS*	AFMB004				
AFMB007	OURSYS*	AFMB006				
AFMB009	OURSYS*	AFMB008				
AFMB012	OURSYS*	AFMB010	AFMB011			
AFMB016	OURSYS*	AFMB013	AFMB014	AFMB015		
AFMB020	OURSYS*	AFMB017	AFMB018	AFMB019		
AFMB024	OURSYS*	AFMB021	AFMB022	AFMB023		
AFMB028	OURSYS*	AFMB025	AFMB026	AFMB027		
AFMB032	OURSYS*	AFMB013	AFMB014	AFMB015		
AFMB036	OURSYS*	AFMB033	AFMB034	AFMB035		
AFMB040	OURSYS*	AFMB037	AFMB038	AFMB039		
AFMB044	OURSYS*	AFMB041	AFMB042	AFMB043		
AF0C301	OURSYS*	OURDMP*	AF0C100	AF0C101	AF0C300	AF0P000
AF0C302	OURSYS*	OURDMP*	AF0C100	AF0C101	AF0C300	AF0P000
AF0C304	OURSYS*	OURDMP*	AF0C102	AF0C103	AF0C303	AF0P000
AF0C305	OURSYS*	OURDMP*	AF0C102	AF0P000		
AF0C306	OURSYS*	OURDMP*	AF0C102	AF0P000		
AF0C307	OURSYS*	OURDMP*	AF0C102	AF0P000		
AFDC311	OURSYS*	OURDMP*	AF0P000	AFDC104	AFDC105	AFDC310
AFDC314	OURSYS*	OURDMP*	AF0P000	AFDC106	<u>AFDC107**</u>	AFDC313
AFDD600	OURSPC*	OURSYS*	AF0P000			
AFDD601	OURSPC*	OURSYS*	AF0P000			
AFDD602	OURSPC*	OURSYS*	AF0P000			
AFDD603	OURSPC*	OURSYS*	AF0P000			
AFDD604	OURSPC*	OURSYS*	AF0P000			
AFDD605	OURSPC*	OURSYS*	AF0P000			
AFDD606	OURSPC*	OURSYS*	AF0P000			
AFDD607	OURSPC*	OURSYS*	AF0P000			
AFDD608*	OURSPC*	OURSYS*	AF0P000			
AFDD609	OURSPC*	OURSYS*	AF0P000			
AFDD610	OURSPC*	OURSYS*	AF0P000			
AFDD611	OURSPC*	OURSYS*	AF0P000			
AF0D720	OURSYS*					
AF0D721	OURSYS*					
AFDD722	OURSYS*	OURTYP*				
AFDD723	OURSYS*	OURTYP*				
AFDD724	OURSYS*	OURTYP*				
AFDD725	OURSYS*	OURTYP*				
AF0D727	OURSYS*					
AF0D728	OURSPC*	OURSYS*				
AF0D729	OURSPC*	OURSYS*				

** Multiple versions of source file supplied(USE and ADA file types)

* Source file supplied with DEC file type only

* Source file supplied with USE file type only

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
AFDD72A	OURSPC*	OURSYS*				
AFDD72B	OURSPC*	OURSYS*				
AFDD72C	OURSPC*	OURSYS*				
AFDD72D	OURSPC*	OURSYS*				
AFDD72E	OURSPC*	OURSYS*				
AFDD72F	OURSPC*	OURSYS*				
AFDD72G	OURSPC*	OURSYS*				
AFDD72H	OURSPC*	OURSYS*				
AFDD72I	OURSPC*	OURSYS*				
AFDDA01	OURSPC*	OURSYS*	AFOP000			
AFDDA02	OURSPC*	OURSYS*	AFOP000			
AFDE220	OURSYS*	OURDMP*	AFOP000	AFDE000	AFDE201	AFDE202
	AFDE203	AFDE204	AFDE205	AFDE206	AFDE207	AFDE208
	AFDE209	AFDE20A	AFDE211	AFDE212	AFDE213	AFDE214
	AFDE215	AFDE216				
AFDE250	OURSYS*	OURDMP*	AFOP000	AFDE000	AFDE231	AFDE232
	AFDE233	AFDE234	AFDE235	AFDE236	AFDE237	AFDE238
	AFDE239	AFDE23A	AFDE241	AFDE242	AFDE243	AFDE244
	AFDE245	AFDE246				
AFDE420	OURSYS*	OURDMP*	AFOP000	AFDE000	AFDE401	AFDE402
	AFDE403	AFDE404	AFDE405	AFDE406	AFDE407	AFDE408
	AFDE409	AFDE40A	AFDE411	AFDE412	AFDE413	AFDE414
	AFDE415	AFDE416				
AFDE450	OURSYS*	OURDMP*	AFOP000	AFDE000	AFDE431	AFDE432
	AFDE433	AFDE434	AFDE435	AFDE436	AFDE437	AFDE438
	AFDE439	AFDE43A	AFDE441	AFDE442	AFDE443	AFDE444
	AFDE445	AFDE446				
AFDF000	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AFDF001	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AFDF002	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AFDF003	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AFDF004	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AFDF005	OURSYS*	OURDMP*	MATHFUN*	AFOP000		
AGD0001	OURSYS*	MATHFUN*	AGOP000	AGD0000		
AGD0003	OURSYS*	AGOP000	AGD0002			
AGD0009	OURSYS*	AGOP000	AGD0004	AGD0005	AGD0006	AGD0007
	AGD0008					
AG0000C	OURSYS*	OURDMP*	AGOP000			
AGD000E	OURSYS*	MATHFUN*	AGOP000	AGD0000	AGD000D	
AGN000K	OURSYS*	AGN000F	AGN000G	AGN000H	AGN000I	AGN000J
AGD0039	OURSPC*	OURSYS*	AGOP000	AGD0004	AGD0005	AGD0006
	AGD0007	AGD0008	AGD0019	AGD0029		
AGD0049	OURSPC*	OURSYS*	AGOP000	AGD0004	AGD0005	AGD0006
	AGD0007	AGD0008	AGD0019	AGD0029		
AGD0059**	OURSPC*	OURSYS*	AGOP000	AGD0004	AGD0005	AGD0006
	AGD0007	AGD0008	AGD0019	AGD0029		
AL09101	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	
AL09111	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	
AL09121	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	
AL09131	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	
ALD9202	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9203	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9204**	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9212	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9213	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9214**	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9222	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9223	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9224**	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9232	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9233	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALD9234**	OURSPC*	OURSYS*	AL09000	AL09100	ALOP000	ALD9200
ALDE2C1	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B1	
ALDE2C2	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B1	

** Source file supplied with DEC file type only

Table G-1 Ada Type A Test Programs and Source Code Files (continued)

PROGRAM		SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)				
ALDE2C3**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B1	
ALDE2C4**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B1	
ALDE2C5	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B2	
ALDE2C6	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B2	
ALDE2C7**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B2	
ALDE2C8**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE2B2	
ALDE4C1	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B1	
ALDE4C2	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B1	
ALDE4C3**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B1	
ALDE4C4**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B1	
ALDE4C5	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B2	
ALDE4C6	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B2	
ALDE4C7**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B2	
ALDE4C8**	OURSPC*	OURSYS*	ALOP000	ALDE000	ALDE4B2	
A000104	OURSYS*	OURDMP*	A000002	A000102	A000103	A00P000
A000109	OURSYS*	OURDMP*	A000001	A000107	A000108	A00P000
A000114	OURSYS*	OURDMP*	A000001	A000112	A000113	A00P000
A000204	OURSYS*	OURDMP*	A000202	A000203	A00P000	
A000209	OURSYS*	OURDMP*	A000001	A000207	A000208	A00P000
A000300	OURSYS*	A00P000				
A000305	OURSYS*	A00P000				
A000310	OURSYS*	A00P000				
A000313	OURSYS*	OURDMP*	A000311	A000312	A00P000	
A000316	OURSYS*	OURDMP*	A000314	A000315	A00P000	
A000319	OURSYS*	OURDMP*	A000317	A000318	A00P000	
A000322	OURSYS*	OURDMP*	A000320	A000321	A00P000	
A000325	OURSYS*	OURDMP*	A000323	A000324	A00P000	
A000328	OURSYS*	OURDMP*	A000326	A000327	A00P000	
A000331	OURSYS*	OURDMP*	A000329	A000330	A00P000	
A000334	OURSYS*	OURDMP*	A000332	A000333	A00P000	
A000337	OURSYS*	OURDMP*	A000335	A000336	A00P000	
A000340	OURSYS*	OURDMP*	A000338	A000339	A00P000	
A000343	OURSYS*	OURDMP*	A000341	A000342	A00P000	
A000346	OURSYS*	OURDMP*	MATHFUN*	A00P000	A000344	A000345
A000350	OURSYS*	OURDMP*	A000347	A000348	A000349	A00P000
A000353	OURSYS*	OURDMP*	A000347	A000351	A000352	A00P000
A000356	OURSYS*	OURDMP*	A000354	A000355	A00P000	
A000359	OURSYS*	OURDMP*	A000357	A000358	A00P000	
A000362	OURSYS*	OURDMP*	A000360	A000361	A00P000	
A000365	OURSPC*	OURSYS*	OURDMP*	A000363	A000364	A00P000
A000368	OURSPC*	OURSYS*	OURDMP*	A000366	A000367	A00P000
A00036B	OURSPC*	OURSYS*	OURDMP*	A000369	A00036A	A00P000
A00036E	OURSPC*	OURSYS*	OURDMP*	A00P000	A00036C	A00036D
A00036H	OURSPC*	OURSYS*	OURDMP*	A00036F	A00036G	A00P000
A00036K	OURSPC*	OURSYS*	OURDMP*	A00036I	A00036J	A00P000
A00036N	OURSPC*	OURSYS*	OURDMP*	A00036L	A00036M	A00P000
A00036Q	OURSPC*	OURSYS*	OURDMP*	A00P000	A00036O	A00036P
A00036T	OURSPC*	OURSYS*	OURDMP*	A00P000	A00036R	A00036S
A00036W	OURSPC*	OURSYS*	OURDMP*	A00036U	A00036V	A00P000
A00036Z	OURSYS*	OURDMP*	OURTYP*	A00P000	A00036X	A00036Y
A000373	OURSPC*	OURSYS*	OURDMP*	A00P000	A000370	A000371
	A000372					
A000378	OURSYS*	OURDMP*	A000374	A000375	A000376	A000377
	A00P000					
A000383	OURSPC*	OURSYS*	OURDMP*	A000374	A00P000	A000380
	A000381	A000382				
A000391	OURSYS*	OURDMP*	A000390	A00P000		
A000404	OURSYS*	OURDMP*	A000402	A000403	A00P000	
A000500	OURSYS*	A00P000				
A000504	OURSYS*	OURDMP*	A000502	A000503	A00P000	
A000509	OURSYS*	OURDMP*	A000507	A000508	A00P000	
A000514	OURSYS*	OURDMP*	A000512	A000513	A00P000	
A000519	OURSYS*	OURDMP*	A000517	A000518	A00P000	
A000525	OURSYS*	OURDMP*	A00P000	A000523	A000524	

** Source file supplied with DEC file type only

Table G-1 Ada Type A Test Programs and Source Code Files (concluded)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
A0D0529	OURSYS*	OURDMP*	A00P000	A0D0526	A0D0527	A0D0528
A000604	OURSYS*	OURDMP*	A000602	A000603	A00P000	
A000605	OURSYS*	OURDMP*	A00P000			
A000606	OURSYS*	OURDMP*	A00P000			
A000607	OURSYS*	OURDMP*	A00P000			
A000704	OURSYS*	OURDMP*	A000702	A000703	A00P000	
A000705	OURSYS*	OURDMP*	A00P000			
A000708	OURSYS*	OURDMP*	A000706	A000707	A00P000	
A000711	OURSYS*	OURDMP*	A000709	A000710	A00P000	
A000713	OURSYS*	OURDMP*	A00P000			
A000714	OURSYS*	OURDMP*	A00P000			
A000715	OURSYS*	OURDMP*	A00P000			
A000716	OURSYS*	OURDMP*	A00P000			
A000717	OURSYS*	OURDMP*	A00P000			
A000718	OURSYS*	OURDMP*	A00P000			
A000721	OURSYS*	OURDMP*	A000719	A000720	A00P000	

Table G-2 Ada Type C Test Programs and Source Code Files

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
CA00000	OURSYS*	OURDMP*	CA0P000			
CF03519	OURSYS*	OURDMP*	CF03500	CF03501	CF03502	CF03503
	CF03504	CF03509	CF03510	CF03511	CF03512	CF03513
	CF03514	CF03517	CF03518	CF0P000		
CF03550	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD3551	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF03620	OURSYS*	OURDMP*	CF03600	CF03601	CF03602	CF03603
	CF03604	CF03605	CF03606	CF03607	CF03608	CF03609
	CF03610	CF03611	CF03612	CF03613	CF03614	CF03615
	CF03616	CF03617	CF03618	CF03619	CF0P000	
CF03650	OURSYS*	OURDMP*	CF03630	CF03631	CF03632	CF03633
	CF03634	CF03635	CF03636	CF03637	CF03638	CF03639
	CF03641	CF03642	CF03643	CF03644	CF03645	CF03646
	CF03647	CF03648	CF03649	CF0P000		
CF03704	OURSPC*	OURSYS*	OURDMP*	CF03700	CF03701	CF03702
	CF03703	CF0P000				
CF03805	OURSPC*	OURSYS*	OURDMP*	CF03800	CF03801	CF03802
	CF03803	CF03804	CF0P000			
CF04120	OURSYS*	OURDMP*	CF0P000			
CF04121	OURSYS*	OURDMP*	CF0P000			
CF04122	OURSYS*	OURDMP*	CF0P000			
CF04123	OURSYS*	OURDMP*	CF0P000			
CF04124	OURSYS*	OURDMP*	CF0P000			
CF04125	OURSYS*	OURDMP*	CF0P000			
CF04126	OURSYS*	OURDMP*	CF0P000			
CF04127	OURSYS*	OURDMP*	CF0P000			
CFD4128*	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04129	OURSYS*	OURDMP*	CF0P000			
CFD412A	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0412B	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD412C	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD412D	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04130	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04131	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04132	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04133	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD4135	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD4136	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD4137	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD4138	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD4139	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD413A	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD413B	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD413C	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0413D	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD413E	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CFD413G	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0413H	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0413I	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0413J	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0413K	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04310	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04311	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04312	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04510	OURSYS*	OURDMP*	CF0P000			
CF04511	OURSYS*	OURDMP*	CF0P000			
CF04512	OURSYS*	OURDMP*	CF0P000			
CF04513	OURSYS*	OURDMP*	CF0P000			
CF04514	OURSYS*	OURDMP*	CF0P000			
CF04515	OURSYS*	OURDMP*	CF0P000			
CF04516	OURSYS*	OURDMP*	CF0P000			
CF04517	OURSYS*	OURDMP*	CF0P000			
CF0451J	OURSYS*	OURDMP*	CF04518	CF04519	CF0451A	CF0451B
	CF0451C	CF0451D	CF0451E	CF0451F	CF0451G	CF0451H
	CF0451I	CF0P000				

* Source file supplied with USE file type only

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
CFD451U*	OURSYS*	OURDMP*	OURTYP*	CF0P000	CFD451K	CFD451L
	CFD451M	CFD451N	CFD451O	CFD451P	CFD451Q	CFD451R
	CFD451S					
CF0451Z	OURSYS*	OURDMP*	CF0451V	CF0451W	CF0451X	CF0451Y
	CF0P000					
CF04520	OURSYS*	OURDMP*	CF0P000			
CF04521	OURSYS*	OURDMP*	CF0P000			
CF04522	OURSYS*	OURDMP*	CF0P000			
CFD4523	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4524	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4525	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4526	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CF04527	OURSYS*	OURDMP*	CF0P000			
CFD4528	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04529	OURSYS*	OURDMP*	CF0P000			
CFD452A	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF04530	OURSYS*	OURDMP*	CF0P000			
CF04531	OURSYS*	OURDMP*	CF0P000			
CF04532	OURSYS*	OURDMP*	CF0P000			
CF04533	OURSYS*	OURDMP*	CF0P000			
CF04534	OURSYS*	OURDMP*	CF0P000			
CF04535	OURSYS*	OURDMP*	CF0P000			
CF04536	OURSYS*	OURDMP*	CF0P000			
CF04537	OURSYS*	OURDMP*	CF0P000			
CF04538	OURSYS*	OURDMP*	CF0P000			
CF04539	OURSYS*	OURDMP*	CF0P000			
CF0453A	OURSYS*	OURDMP*	CF0P000			
CF0453B	OURSYS*	OURDMP*	CF0P000			
CF0453C	OURSYS*	OURDMP*	CF0P000			
CF04540	OURSYS*	OURDMP*	CF0P000			
CF04541	OURSYS*	OURDMP*	CF0P000			
CF04550	OURSYS*	OURDMP*	CF0P000			
CF04551	OURSYS*	OURDMP*	CF0P000			
CF04552	OURSYS*	OURDMP*	CF0P000			
CF04553	OURSYS*	OURDMP*	CF0P000			
CF04554	OURSYS*	OURDMP*	CF0P000			
CF04555	OURSYS*	OURDMP*	CF0P000			
CF04556	OURSYS*	OURDMP*	CF0P000			
CF04557	OURSYS*	OURDMP*	CF0P000			
CF04558	OURSYS*	OURDMP*	CF0P000			
CF04559	OURSYS*	OURDMP*	CF0P000			
CFD455D	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455E	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455F	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455G	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455I	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455J	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455K	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD455L	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CF0455N	OURSYS*	OURDMP*	CF0P000			
CF0455O	OURSYS*	OURDMP*	CF0P000			
CF04560	OURSYS*	OURDMP*	CF0P000			
CF04562	OURSYS*	OURDMP*	CF0P000			
CF04563	OURSYS*	OURDMP*	CF0P000			
CFD4566	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4567	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4568	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4569	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CF04600	OURSYS*	OURDMP*	CF0P000			
CF04601	OURSYS*	OURDMP*	CF0P000			
CF04602	OURSYS*	OURDMP*	CF0P000			
CFD4603	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4604	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4605	OURSYS*	OURDMP*	OURTYP*	CF0P000		
CFD4606	OURSYS*	OURDMP*	OURTYP*	CF0P000		

* Source file supplied with USE file type only

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
CFD4607	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD4608	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD4609	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD460A	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD460B	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CF0460E	OURSYS*	OURDMP*	CFOP000			
CF0460F	OURSYS*	OURDMP*	CFOP000			
CF0460G	OURSYS*	OURDMP*	CFOP000			
CF0460H	OURSYS*	OURDMP*	CFOP000			
CF0460I	OURSYS*	OURDMP*	CFOP000			
CF0460J	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD460K	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD460L	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CF0460M	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD460N	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD4600	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD4804	OURSPC*	OURSYS*	OURDMP*	CFOP000	CFD4800	CFD4801
	CFD4802	CFD4803**				
CF05200	OURSYS*	OURDMP*	CFOP000			
CF05201	OURSYS*	OURDMP*	CFOP000			
CF05202	OURSYS*	OURDMP*	CFOP000			
CF05203	OURSYS*	OURDMP*	CFOP000			
CF05204	OURSYS*	OURDMP*	CFOP000			
CF05205	OURSYS*	OURDMP*	CFOP000			
CF05206	OURSYS*	OURDMP*	CFOP000			
CF05207	OURSYS*	OURDMP*	CFOP000			
CF05208	OURSYS*	OURDMP*	CFOP000			
CF05209	OURSYS*	OURDMP*	CFOP000			
CF0520A	OURSYS*	OURDMP*	CFOP000			
CF0520B	OURSYS*	OURDMP*	CFOP000			
CFD520C	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520D	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520E	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520F	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520G	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520H	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520I	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD520J	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CF0520M	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD520N	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CF05200	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD520P	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CFD520Q	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CF0520R	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CF0520S	OURSPC*	OURSYS*	OURDMP*	CFOP000		
CF05210	OURSYS*	OURDMP*	CFOP000			
CF05211	OURSYS*	OURDMP*	CFOP000			
CF05212	OURSYS*	OURDMP*	CFOP000			
CF05213	OURSYS*	OURDMP*	CFOP000			
CF05214	OURSYS*	OURDMP*	CFOP000			
CF05215	OURSYS*	OURDMP*	CFOP000			
CFD5218	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD5219	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD521A	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD521B	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD521C	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD521D	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CFD521E	OURSYS*	OURDMP*	OURTYP*	CFOP000		
CF0521H	OURSYS*	OURDMP*	CFOP000			
CF0521I	OURSYS*	OURDMP*	CFOP000			
CF0521J	OURSYS*	OURDMP*	CFOP000			
CF0521K	OURSYS*	OURDMP*	CFOP000			
CFD521K	OURSPC*	OURSYS*	OURDMP*	CFOP000		

** Multiple versions of source file supplied(USE and ADA file types)

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
CF0521L	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0521M	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0521N	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0521O	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF0521P	OURSPC*	OURSYS*	OURDMP*	CF0P000		
CF05304	OURSYS*	OURDMP*	CF05300	CF05301	CF05302	CF05303
	CF0P000					
CF05308	OURSYS*	OURDMP*	CF05305	CF05306	CF05307	CF0P000
CF05408	OURSYS*	OURDMP*	CF05400	CF05401	CF05402	CF05403
	CF05404	CF05405	CF05406	CF05407	CF0P000	
CF05505	OURSYS*	OURDMP*	CF05501	CF05502	CF05503	CF05504
	CF0P000					
CF0550C	OURSYS*	OURDMP*	CF05506	CF05507	CF05508	CF05509
	CF0550A	CF0550B	CF0P000			
CF0550D	OURSYS*	OURDMP*	CF0P000			
CF0550E	OURSYS*	OURDMP*	CF0P000			
CF0550F	OURSYS*	OURDMP*	CF0P000			
CF06001	OURSYS*	OURDMP*	CF0P000			
CF06009	OURSYS*	OURDMP*	CF0P000			
CF06010	OURSYS*	OURDMP*	CF0P000			
CF06011	OURSYS*	OURDMP*	CF0P000			
CF06022	OURSYS*	OURDMP*	CF06013	CF06014	CF06015	CF06016
	CF06017	CF06018	CF06019	CF0P000		
CF06033	OURSYS*	OURDMP*	CF06023	CF06024	CF06025	CF06026
	CF06027	CF06028	CF06029	CF06030	CF06031	CF06032
	CF0P000					
CF06053	OURSYS*	OURDMP*	CF06043	CF06044	CF06045	CF06046
	CF06047	CF06048	CF06049	CF06050	CF06051	CF06052
	CF0P000					
CF06069	OURSYS*	OURDMP*	CF06060	CF06061	CF06062	CF06063
	CF06064	CF06065	CF06066	CF06067	CF06068	CF0P000
CF06079	OURSYS*	OURDMP*	CF06070	CF06071	CF06072	CF06073
	CF06074	CF06075	CF06076	CF06077	CF06078	CF0P000
CF06101	OURSYS*	OURDMP*	CF06100	CF0P000		
CF06109	OURSYS*	OURDMP*	CF06108	CF0P000		
CF06110	OURSYS*	OURDMP*	CF06108	CF0P000		
CF06111	OURSYS*	OURDMP*	CF06108	CF0P000		
CF06122	OURSYS*	OURDMP*	CF06112	CF06113	CF06114	CF06115
	CF06116	CF06117	CF06118	CF06119	CF0P000	
CF06140	OURSYS*	OURDMP*	CF06132	CF06133	CF06134	CF06135
	CF06136	CF06137	CF06138	CF06139	CF0P000	
CF06150	OURSYS*	OURDMP*	CF06142	CF06143	CF06144	CF06145
	CF06146	CF06147	CF06148	CF06149	CF0P000	
CF06160	OURSYS*	OURDMP*	CF06152	CF06153	CF06154	CF06155
	CF06156	CF06157	CF06158	CF06159	CF0P000	
CF06170	OURSYS*	OURDMP*	CF06162	CF06163	CF06164	CF06165
	CF06166	CF06167	CF06168	CF06169	CF0P000	
CF06180	OURSYS*	OURDMP*	CF06172	CF06173	CF06174	CF06175
	CF06176	CF06177	CF06178	CF06179	CF0P000	
CF06190	OURSYS*	OURDMP*	CF06182	CF06183	CF06184	CF06185
	CF06186	CF06187	CF06188	CF06189	CF0P000	
CF06199	OURSPC*	OURSYS*	OURDMP*	CF06191	CF06192	CF06193
	CF06194	CF06195	CF06196	CF06197	CF06198	CF0P000
CF0619I	OURSPC*	OURSYS*	OURDMP*	CF0619A	CF0619B	CF0619C
	CF0619D	CF0619E	CF0619F	CF0619G	CF0619H	CF0P000
CF0619R	OURSPC*	OURSYS*	OURDMP*	CF0619J	CF0619K	CF0619L
	CF0619M	CF0619N	CF0619O	CF0619P	CF0619Q	CF0P000
CFD6201	OURSYS*	OURDMP*	CF0P000			
CFD6209	OURSYS*	OURDMP*	CF0P000			
CFD6210	OURSYS*	OURDMP*	CF0P000			
CFD6211	OURSYS*	OURDMP*	CF0P000			
CFD6222	OURSYS*	OURDMP*	CF0P000	CFD6213	CFD6214	CFD6215
	CFD6216	CFD6217	CFD6218	CFD6219		
CFD6233	OURSYS*	OURDMP*	CF0P000	CFD6223	CFD6224	CFD6225
	CFD6226	CFD6227	CFD6228	CFD6229	CFD6230	CFD6231

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
CFD6253	CFD6232 OURSYSX CFD6246 CFD6252	OURDMPX CFD6247	CF0P000 CFD6248	CFD6243 CFD6249	CFD6244 CFD6250	CFD6245 CFD6251
CFD6269	OURSYSX CFD6263	OURDMPX CFD6264	CF0P000 CFD6265	CFD6260 CFD6266	CFD6261 CFD6267	CFD6262 CFD6268
CFD6279	OURSYSX CFD6273	OURDMPX CFD6274	CF0P000 CFD6275	CFD6270 CFD6276	CFD6271 CFD6277	CFD6272 CFD6278
CFD6301	OURSYSX	OURDMPX	CF0P000	CFD6300		
CFD6309	OURSYSX	OURDMPX	CF0P000	CFD6308		
CFD6310	OURSYSX	OURDMPX	CF0P000	CFD6308		
CFD6311	OURSYSX	OURDMPX	CF0P000	CFD6308		
CFD6322	OURSYSX CFD6315	OURDMPX CFD6316	CF0P000 CFD6317	CFD6312 CFD6318	CFD6313 CFD6319	CFD6314
CFD6340	OURSYSX CFD6335	OURDMPX CFD6336	CF0P000 CFD6337	CFD6332 CFD6338	CFD6333 CFD6339	CFD6334
CFD6350	OURSYSX CFD6345	OURDMPX CFD6346	CF0P000 CFD6347	CFD6342 CFD6348	CFD6343 CFD6349	CFD6344
CFD6360	OURSYSX CFD6355	OURDMPX CFD6356	CF0P000 CFD6357	CFD6352 CFD6358	CFD6353 CFD6359	CFD6354
CFD6370	OURSYSX CFD6365	OURDMPX CFD6366	CF0P000 CFD6367	CFD6362 CFD6368	CFD6363 CFD6369	CFD6364
CFD6380	OURSYSX CFD6375	OURDMPX CFD6376	CF0P000 CFD6377	CFD6372 CFD6378	CFD6373 CFD6379	CFD6374
CFD6390	OURSYSX CFD6385	OURDMPX CFD6386	CF0P000 CFD6387	CFD6382 CFD6388	CFD6383 CFD6389	CFD6384
CF0642B	OURSPCX CF06425 CF0P000	OURSYSX CF06426	OURDMPX CF06427	CF06422 CF06428	CF06423 CF06429	CF06424 CF0642A
CF06803	OURSYSX	OURDMPX	CF06802	CF0P000		
CF06807	OURSYSX	OURDMPX	CF06806	CF0P000		
CF06809	OURSPCX	OURSYSX	OURDMPX	CF06808	CF0P000	
CF0680B	OURSPCX	OURSYSX	OURDMPX	CF0680A	CF0P000	
CF0680D	OURSPCX	OURSYSX	OURDMPX	CF0680C	CF0P000	
CF0680F	OURSPCX	OURSYSX	OURDMPX	CF0680E	CF0P000	
CFD680H	OURSPCX	OURSYSX	OURDMPX	CF0P000	CFD680G	
CFD680J	OURSPCX	OURSYSX	OURDMPX	CF0P000	CFD680I	
CF06811	OURSYSX	OURDMPX	CF06810	CF0P000		
CF06815	OURSYSX	OURDMPX	CF06814	CF0P000		
CF06817	OURSYSX	OURDMPX	CF06816	CF0P000		
CF06819	OURSYSX	OURDMPX	CF06818	CF0P000		
CF06821	OURSYSX	OURDMPX	CF06820	CF0P000		
CF06823	OURSPCX	OURSYSX	OURDMPX	CF06822	CF0P000	
CF06825	OURSPCX	OURSYSX	OURDMPX	CF06824	CF0P000	
CFN9301	OURSPCX	OURSYSX	CFN9300			
CFN9302	OURSPCX	OURSYSX	CFN9300			
CFN9303	OURSPCX	OURSYSX	CFN9300			
CF09501	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09502	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09503	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09504	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09505	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09506	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09507	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09508	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CF09509	OURSPCX	OURSYSX	OURDMPX	CF09500	CF0P000	
CFN9511	OURSPCX	OURSYSX	CFN9510			
CF09600	OURSPCX	OURSYSX	CF0P000			
CF09601	OURSPCX	OURSYSX	CF0P000			
CF09602	OURSPCX	OURSYSX	CF0P000			
CF09603	OURSPCX	OURSYSX	CF0P000			
CF09604	OURSPCX	OURSYSX	CF0P000			
CF09605	OURSPCX	OURSYSX	CF0P000			
CF09606	OURSPCX	OURSYSX	CF0P000			
CF09607	OURSPCX	OURSYSX	CF0P000			

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
CFN9611	OURSPC*	OURSYS*	CFN9610			
CFN9612	OURSPC*	OURSYS*	CFN9610			
CFN9613	OURSPC*	OURSYS*	CFN9610			
CF09711	OURSYS*	CF09710	CF0P000			
CF09713	OURSYS*	CF09712	CF0P000			
CF09720	OURSPC*	OURSYS*	OURDMP*	CF09500	CF0P000	
CF09721	OURSPC*	OURSYS*	OURDMP*	CF09500	CF0P000	
CF09731	OURSPC*	OURSYS*	OURDMP*	CF09500	CF0P000	
CF09901	OURSYS*	CF09900	CF0P000			
CF09902	OURSYS*	CF09900	CF0P000			
CF09903	OURSYS*	CF09900	CF0P000			
CFM9A01	OURSPC*	OURSYS*	CFM9A00			
CFM9A02	OURSPC*	OURSYS*	CFM9A00			
CFM9A03	OURSPC*	OURSYS*	CFM9A00			
CF09B01	OURSPC*	OURSYS*	OURDMP*	CF09500	CF0P000	
CFD9C00	OURSPC*	OURSYS*	CF0P000	CFD9000		
CFD9C01	OURSPC*	OURSYS*	CF0P000	CFD9000	CFD9200	
CFD9C02	OURSPC*	OURSYS*	CF0P000	CFD9000	CFD9200	
CFD9C03**	OURSPC*	OURSYS*	CF0P000	CFD9000	CFD9200	
CFD9C04**	OURSPC*	OURSYS*	CF0P000	CFD9000	CFD9200	
CFMB001	OURSYS*	CFMB000				
CFMB003	OURSYS*	CFMB002				
CFMB005	OURSYS*	CFMB004				
CFMB012	OURSYS*	CFMB010	CFMB011			
CFMB016	OURSYS*	CFMB013	CFMB014	CFMB015		
CFMB020	OURSYS*	CFMB017	CFMB018	CFMB019		
CFMB032	OURSYS*	CFMB013	CFMB014	CFMB015		
CFMB036	OURSYS*	CFMB033	CFMB034	CFMB035		
CFMB044	OURSYS*	CFMB041	CFMB042	CFMB043		
CF0C301	OURSYS*	OURDMP*	CF0C100	CF0C101	CF0C300	CF0P000
CF0C302	OURSYS*	OURDMP*	CF0C100	CF0C101	CF0C300	CF0P000
CF0C304	OURSYS*	OURDMP*	CF0C102	CF0C103	CF0C303	CF0P000
CF0C305	OURSYS*	OURDMP*	CF0C102	CF0P000		
CF0C306	OURSYS*	OURDMP*	CF0C102	CF0P000		
CF0C307	OURSYS*	OURDMP*	CF0C102	CF0P000		
CFDC311	OURSYS*	OURDMP*	CF0P000	CFDC104	CFDC105	CFDC310
CFDC314	OURSYS*	OURDMP*	CF0P000	CFDC106	CFDC107**	CFDC313
CFDD600	OURSPC*	OURSYS*	CF0P000			
CFDD601	OURSPC*	OURSYS*	CF0P000			
CFDD602	OURSPC*	OURSYS*	CF0P000			
CFDD603	OURSPC*	OURSYS*	CF0P000			
CFDD604	OURSPC*	OURSYS*	CF0P000			
CFDD605	OURSPC*	OURSYS*	CF0P000			
CFDD606	OURSPC*	OURSYS*	CF0P000			
CFDD607	OURSPC*	OURSYS*	CF0P000			
CFDD608*	OURSPC*	OURSYS*	CF0P000			
CFDD609	OURSPC*	OURSYS*	CF0P000			
CFDD610	OURSPC*	OURSYS*	CF0P000			
CFDD611	OURSPC*	OURSYS*	CF0P000			
CFDDA01	OURSPC*	OURSYS*	CF0P000			
CFDDA02	OURSPC*	OURSYS*	CF0P000			
CFDE220	OURSYS*	OURDMP*	CF0P000	CFDE000	CFDE201	CFDE202
	CFDE203	CFDE204	CFDE205	CFDE206	CFDE207	CFDE208
	CFDE209	CFDE20A	CFDE211	CFDE212	CFDE213	CFDE214
	CFDE215	CFDE216				
CFDE250	OURSYS*	OURDMP*	CF0P000	CFDE000	CFDE231	CFDE232
	CFDE233	CFDE234	CFDE235	CFDE236	CFDE237	CFDE238
	CFDE239	CFDE23A	CFDE241	CFDE242	CFDE243	CFDE244
	CFDE245	CFDE246				
CFDE420	OURSYS*	OURDMP*	CF0P000	CFDE000	CFDE401	CFDE402
	CFDE403	CFDE404	CFDE405	CFDE406	CFDE407	CFDE408
	CFDE409	CFDE40A	CFDE411	CFDE412	CFDE413	CFDE414
	CFDE415	CFDE416				

** Multiple versions of source file supplied(USE and ADA file types)
 ** Source file supplied with DEC file type only
 * Source file supplied with USE file type only

Table G-2 Ada Type C Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
CFDE450	OURSYS*	OURDMP*	CFOP000	CFDE000	CFDE431	CFDE432
	CFDE433	CFDE434	CFDE435	CFDE436	CFDE437	CFDE438
	CFDE439	CFDE43A	CFDE441	CFDE442	CFDE443	CFDE444
	CFDE445	CFDE446				
CFDF000	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CFDF001	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CFDF002	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CFDF003	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CFDF004	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CFDF005	OURSYS*	OURDMP*	MATHFUN*	CFOP000		
CGD0001	OURSYS*	MATHFUN*	CGOP000	CGD0000		
CGD0003	OURSYS*	CGOP000	CGD0002			
CGD0009	OURSYS*	CGOP000	CGD0004	CGD0005	CGD0006	CGD0007
	CGD0008					
CG0000C	OURSYS*	OURDMP*	CGOP000			
CGD000E	OURSYS*	MATHFUN*	CGOP000	CGD0000	CGD000D	
CGN000K	OURSYS*	CGN000F	CGN000G	CGN000H	CGN000I	CGN000J
CGD0039	OURSPC*	OURSYS*	CGOP000	CGD0004	CGD0005	CGD0006
	CGD0007	CGD0008	CGD0019	CGD0029		
CGD0049	OURSPC*	OURSYS*	CGOP000	CGD0004	CGD0005	CGD0006
	CGD0007	CGD0008	CGD0019	CGD0029		
CGD0059**	OURSPC*	OURSYS*	CGOP000	CGD0004	CGD0005	CGD0006
	CGD0007	CGD0008	CGD0019	CGD0029		
CL09101	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	
CL09111	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	
CL09121	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	
CL09131	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	
CLD9202	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9203	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9204**	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9212	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9213	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9214**	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9222	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9223	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9224**	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9232	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9233	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLD9234**	OURSPC*	OURSYS*	CL09000	CL09100	CL0P000	CLD9200
CLDE2C1	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B1	
CLDE2C2	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B1	
CLDE2C3**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B1	
CLDE2C4**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B1	
CLDE2C5	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B2	
CLDE2C6	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B2	
CLDE2C7**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B2	
CLDE2C8**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE2B2	
CLDE4C1	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B1	
CLDE4C2	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B1	
CLDE4C3**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B1	
CLDE4C4**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B1	
CLDE4C5	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B2	
CLDE4C6	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B2	
CLDE4C7**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B2	
CLDE4C8**	OURSPC*	OURSYS*	CL0P000	CLDE000	CLDE4B2	
C000104	OURSYS*	OURDMP*	C000002	C000102	C000103	C00P000
C000109	OURSYS*	OURDMP*	C000001	C000107	C000108	C00P000
C000114	OURSYS*	OURDMP*	C000001	C000112	C000113	C00P000
C000204	OURSYS*	OURDMP*	C000207	C000203	C00P000	
C000209	OURSYS*	OURDMP*	C000001	C000207	C000208	C00P000
C000300	OURSYS*	C00P000				
C000305	OURSYS*	C00P000				
C000310	OURSYS*	C00P000				
C000313	OURSYS*	OURDMP*	C000311	C000312	C00P000	

** Source file supplied with DEC file type only

Table G-2 Ada Type C Test Programs and Source Code Files (concluded)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
C000316	OURSYS*	OURDMP*	C000314	C000315	C00P000	
C000319	OURSYS*	OURDMP*	C000317	C000318	C00P000	
C000322	OURSYS*	OURDMP*	C000320	C000321	C00P000	
C000325	OURSYS*	OURDMP*	C000323	C000324	C00P000	
C000328	OURSYS*	OURDMP*	C000326	C000327	C00P000	
C000331	OURSYS*	OURDMP*	C000329	C000330	C00P000	
C000334	OURSYS*	OURDMP*	C000332	C000333	C00P000	
C000337	OURSYS*	OURDMP*	C000335	C000336	C00P000	
C000340	OURSYS*	OURDMP*	C000338	C000339	C00P000	
C000343	OURSYS*	OURDMP*	C000341	C000342	C00P000	
C000346	OURSYS*	OURDMP*	MATHFUN*	C00P000	C000344	C000345
C000350	OURSYS*	OURDMP*	C000347	C000348	C000349	C00P000
C000353	OURSYS*	OURDMP*	C000347	C000351	C000352	C00P000
C000356	OURSYS*	OURDMP*	C000354	C000355	C00P000	
C000359	OURSYS*	OURDMP*	C000357	C000358	C00P000	
C000362	OURSYS*	OURDMP*	C000360	C000361	C00P000	
C000365	OURSPC*	OURSYS*	OURDMP*	C000363	C000364	C00P000
C000368	OURSPC*	OURSYS*	OURDMP*	C000366	C000367	C00P000
C00036B	OURSPC*	OURSYS*	OURDMP*	C000369	C00036A	C00P000
C00036E	OURSPC*	OURSYS*	OURDMP*	C00P000	C00036C	C00036D
C00036H	OURSPC*	OURSYS*	OURDMP*	C00036F	C00036G	C00P000
C00036K	OURSPC*	OURSYS*	OURDMP*	C00036I	C00036J	C00P000
C00036N	OURSPC*	OURSYS*	OURDMP*	C00036L	C00036M	C00P000
C00036Q	OURSPC*	OURSYS*	OURDMP*	C00P000	C00036O	C00036P
C00036T	OURSPC*	OURSYS*	OURDMP*	C00P000	C00036R	C00036S
C00036W	OURSPC*	OURSYS*	OURDMP*	C00036U	C00036V	C00P000
C00036Z	OURSYS*	OURDMP*	OURTYP*	C00P000	C00036X	C00036Y
C000373	OURSPC*	OURSYS*	OURDMP*	C00P000	C000370	C000371
C000378	C000372	OURDMP*	C000374	C000375	C000376	C000377
C000383	C00P000	OURDMP*	OURDMP*	C000374	C00P000	C000380
C000391	OURSPC*	OURSYS*	C000390	C00P000	C00P000	
C000404	OURSYS*	OURDMP*	C000402	C000403	C00P000	
C000500	OURSYS*	C00P000	C000502	C000503	C00P000	
C000504	OURSYS*	OURDMP*	C000507	C000508	C00P000	
C000509	OURSYS*	OURDMP*	C000512	C000513	C00P000	
C000514	OURSYS*	OURDMP*	C000517	C000518	C00P000	
C000519	OURSYS*	OURDMP*	C00P000	C000523	C000524	
C000525	OURSYS*	OURDMP*	C00P000	C000526	C000527	C000528
C000529	OURSYS*	OURDMP*	C000602	C000603	C00P000	
C000604	OURSYS*	OURDMP*	C00P000			
C000605	OURSYS*	OURDMP*	C00P000			
C000606	OURSYS*	OURDMP*	C00P000			
C000607	OURSYS*	OURDMP*	C00P000			
C000704	OURSYS*	OURDMP*	C000702	C000703	C00P000	
C000705	OURSYS*	OURDMP*	C00P000			
C000708	OURSYS*	OURDMP*	C000706	C000707	C00P000	
C000711	OURSYS*	OURDMP*	C000709	C000710	C00P000	
C000713	OURSYS*	OURDMP*	C00P000			
C000714	OURSYS*	OURDMP*	C00P000			
C000715	OURSYS*	OURDMP*	C00P000			
C000716	OURSYS*	OURDMP*	C00P000			
C000717	OURSYS*	OURDMP*	C00P000			
C000718	OURSYS*	OURDMP*	C00P000			
C000721	OURSYS*	OURDMP*	C000719	C000720	C00P000	

Table 7-3 Ada Type E Test Programs and Source Code Files

PROGRAM	SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)					
EA00000	OURSYSX	OURDMPX	EA0P000			
EF03519	OURSYSX	OURDMPX	EF03500	EF03501	EF03502	EF03503
	EF03504	EF03509	EF03510	EF03511	EF03512	EF03513
	EF03514	EF03517	EF03518	EF0P000		
EF03550	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD3551	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF03620	OURSYSX	OURDMPX	EF03600	EF03601	EF03602	EF03603
	EF03604	EF03605	EF03606	EF03607	EF03608	EF03609
	EF03610	EF03611	EF03612	EF03613	EF03614	EF03615
	EF03616	EF03617	EF03618	EF03619	EF0P000	
EF03650	OURSYSX	OURDMPX	EF03630	EF03631	EF03632	EF03633
	EF03634	EF03635	EF03636	EF03637	EF03638	EF03639
	EF03641	EF03642	EF03643	EF03644	EF03645	EF03646
	EF03647	EF03648	EF03649	EF0P000		
EF03704	OURSPCX	OURSYSX	OURDMPX	EF03700	EF03701	EF03702
	EF03703	EF0P000				
EF03805	OURSPCX	OURSYSX	OURDMPX	EF03800	EF03801	EF03802
	EF03803	EF03804	EF0P000			
EF04120	OURSYSX	OURDMPX	EF0P000			
EF04121	OURSYSX	OURDMPX	EF0P000			
EF04122	OURSYSX	OURDMPX	EF0P000			
EF04123	OURSYSX	OURDMPX	EF0P000			
EF04124	OURSYSX	OURDMPX	EF0P000			
EF04125	OURSYSX	OURDMPX	EF0P000			
EF04126	OURSYSX	OURDMPX	EF0P000			
EF04127	OURSYSX	OURDMPX	EF0P000			
EFD4128*	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04129	OURSYSX	OURDMPX	EF0P000	EF0P000		
EFD412A	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0412B	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD412C	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD412D	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04130	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04131	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04132	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04133	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD4135	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD4136	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD4137	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD4138	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD4139	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD413A	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD413B	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD413C	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0413D	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD413E	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EFD413G	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0413H	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0413I	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0413J	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF0413K	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04310	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04311	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04312	OURSPCX	OURSYSX	OURDMPX	EF0P000		
EF04510	OURSYSX	OURDMPX	EF0P000			
EF04511	OURSYSX	OURDMPX	EF0P000			
EF04512	OURSYSX	OURDMPX	EF0P000			
EF04513	OURSYSX	OURDMPX	EF0P000			
EF04514	OURSYSX	OURDMPX	EF0P000			
EF04515	OURSYSX	OURDMPX	EF0P000			
EF04516	OURSYSX	OURDMPX	EF0P000			
EF04517	OURSYSX	OURDMPX	EF0P000			
EF0451J	OURSYSX	OURDMPX	EF04518	EF04519	EF0451A	EF0451B
	EF0451C	EF0451D	EF0451E	EF0451F	EF0451G	EF0451H
	EF0451I	EF0P000				

* Source file supplied with USE file type only

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
EFD451U*	OURSYS*	OURDMP*	OURTYP*	EF0P000	EFD451K	EFD451L
	EFD451M	EFD451N	EFD451O	EFD451P	EFD451Q	EFD451R
	EFD451S					
EF0451Z	OURSYS*	OURDMP*	EF0451V	EF0451W	EF0451X	EF0451Y
	EF0P000					
EF04520	OURSYS*	OURDMP*	EF0P000			
EF04521	OURSYS*	OURDMP*	EF0P000			
EF04522	OURSYS*	OURDMP*	EF0P000			
EFD4523	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4524	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4525	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4526	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EF04527	OURSYS*	OURDMP*	EF0P000			
EFD4528	OURSPC*	OURSYS*	OURDMP*	EF0P000		
EF04529	OURSYS*	OURDMP*	EF0P000			
EFD452A	OURSPC*	OURSYS*	OURDMP*	EF0P000		
EF04530	OURSYS*	OURDMP*	EF0P000			
EF04531	OURSYS*	OURDMP*	EF0P000			
EF04532	OURSYS*	OURDMP*	EF0P000			
EF04533	OURSYS*	OURDMP*	EF0P000			
EF04534	OURSYS*	OURDMP*	EF0P000			
EF04535	OURSYS*	OURDMP*	EF0P000			
EF04536	OURSYS*	OURDMP*	EF0P000			
EF04537	OURSYS*	OURDMP*	EF0P000			
EF04538	OURSYS*	OURDMP*	EF0P000			
EF04539	OURSYS*	OURDMP*	EF0P000			
EF0453A	OURSYS*	OURDMP*	EF0P000			
EF0453B	OURSYS*	OURDMP*	EF0P000			
EF0453C	OURSYS*	OURDMP*	EF0P000			
EF04540	OURSYS*	OURDMP*	EF0P000			
EF04541	OURSYS*	OURDMP*	EF0P000			
EF04550	OURSYS*	OURDMP*	EF0P000			
EF04551	OURSYS*	OURDMP*	EF0P000			
EF04552	OURSYS*	OURDMP*	EF0P000			
EF04553	OURSYS*	OURDMP*	EF0P000			
EF04554	OURSYS*	OURDMP*	EF0P000			
EF04555	OURSYS*	OURDMP*	EF0P000			
EF04556	OURSYS*	OURDMP*	EF0P000			
EF04557	OURSYS*	OURDMP*	EF0P000			
EF04558	OURSYS*	OURDMP*	EF0P000			
EF04559	OURSYS*	OURDMP*	EF0P000			
EFD455D	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455E	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455F	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455G	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455I	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455J	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455K	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD455L	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EF0455N	OURSYS*	OURDMP*	EF0P000			
EF0455O	OURSYS*	OURDMP*	EF0P000			
EF04560	OURSYS*	OURDMP*	EF0P000			
EF04562	OURSYS*	OURDMP*	EF0P000			
EF04563	OURSYS*	OURDMP*	EF0P000			
EFD4566	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4567	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4568	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4569	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EF04600	OURSYS*	OURDMP*	EF0P000			
EF04601	OURSYS*	OURDMP*	EF0P000			
EF04602	OURSYS*	OURDMP*	EF0P000			
EFD4603	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4604	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4605	OURSYS*	OURDMP*	OURTYP*	EF0P000		
EFD4606	OURSYS*	OURDMP*	OURTYP*	EF0P000		

* Source file supplied with USE file type only

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>			
EFD4607	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD4608	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD4609	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD460A	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD460B	OURSYS*	OURDMP*	OURTYP*	EFOP000
EF0460E	OURSYS*	OURDMP*	EFOP000	
EF0460F	OURSYS*	OURDMP*	EFOP000	
EF0460G	OURSYS*	OURDMP*	EFOP000	
EF0460H	OURSYS*	OURDMP*	EFOP000	
EF0460I	OURSYS*	OURDMP*	EFOP000	
EF0460J	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD460K	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD460L	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0460M	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD460N	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD460O	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF05200	OURSYS*	OURDMP*	EFOP000	
EF05201	OURSYS*	OURDMP*	EFOP000	
EF05202	OURSYS*	OURDMP*	EFOP000	
EF05203	OURSYS*	OURDMP*	EFOP000	
EF05204	OURSYS*	OURDMP*	EFOP000	
EF05205	OURSYS*	OURDMP*	EFOP000	
EF05206	OURSYS*	OURDMP*	EFOP000	
EF05207	OURSYS*	OURDMP*	EFOP000	
EF05208	OURSYS*	OURDMP*	EFOP000	
EF05209	OURSYS*	OURDMP*	EFOP000	
EF0520A	OURSYS*	OURDMP*	EFOP000	
EF0520B	OURSYS*	OURDMP*	EFOP000	
EFD520C	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520D	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520E	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520F	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520G	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520H	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520I	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD520J	OURSYS*	OURDMP*	OURTYP*	EFOP000
EF0520M	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD520N	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0520O	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD520P	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD520Q	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0520R	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0520S	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF05210	OURSYS*	OURDMP*	EFOP000	
EF05211	OURSYS*	OURDMP*	EFOP000	
EF05212	OURSYS*	OURDMP*	EFOP000	
EF05213	OURSYS*	OURDMP*	EFOP000	
EF05214	OURSYS*	OURDMP*	EFOP000	
EF05215	OURSYS*	OURDMP*	EFOP000	
EFD5218	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD5219	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521A	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521B	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521C	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521D	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521E	OURSYS*	OURDMP*	OURTYP*	EFOP000
EFD521F	OURSYS*	OURDMP*	OURTYP*	EFOP000
EF0521H	OURSYS*	OURDMP*	EFOP000	
EF0521I	OURSYS*	OURDMP*	EFOP000	
EF0521J	OURSYS*	OURDMP*	EFOP000	
EF0521K	OURSYS*	OURDMP*	EFOP000	
EFD521K	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0521L	OURSPC*	OURSYS*	OURDMP*	EFOP000
EFD521M	OURSPC*	OURSYS*	OURDMP*	EFOP000
EF0521N	OURSPC*	OURSYS*	OURDMP*	EFOP000

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
EFD5210	OURSPC*	OURSYS*	OURDMP*	EF0P000		
EFD521P	OURSPC*	OURSYS*	OURDMP*	EF0P000		
EF05304	OURSYS*	OURDMP*	EF05300	EF05301	EF05302	EF05303
	EF0P000					
EF05308	OURSYS*	OURDMP*	EF05305	EF05306	EF05307	EF0P000
EF05408	OURDMP*	EF05400	EF05401	EF05402	EF05403	
	EF05404	EF05405	EF05406	EF05407	EF0P000	
EF05505	OURSYS*	OURDMP*	EF05501	EF05502	EF05503	EF05504
	EF0P000					
EF0550C	OURSYS*	OURDMP*	EF05506	EF05507	EF05508	EF05509
	EF0550A	EF0550B	EF0P000			
EF0550D	OURSYS*	OURDMP*	EF0P000			
EF0550E	OURSYS*	OURDMP*	EF0P000			
EF0550F	OURSYS*	OURDMP*	EF0P000			
EF06001	OURSYS*	OURDMP*	EF0P000			
EF06009	OURSYS*	OURDMP*	EF0P000			
EF06010	OURSYS*	OURDMP*	EF0P000			
EF06011	OURSYS*	OURDMP*	EF0P000			
EF06022	OURSYS*	OURDMP*	EF06013	EF06014	EF06015	EF06016
	EF06017	EF06018	EF06019	EF0P000		
EF06033	OURSYS*	OURDMP*	EF06023	EF06024	EF06025	EF06026
	EF06027	EF06028	EF06029	EF06030	EF06031	EF06032
	EF0P000					
EF06053	OURSYS*	OURDMP*	EF06043	EF06044	EF06045	EF06046
	EF06047	EF06048	EF06049	EF06050	EF06051	EF06052
	EF0P000					
EF06069	OURSYS*	OURDMP*	EF06060	EF06061	EF06062	EF06063
	EF06064	EF06065	EF06066	EF06067	EF06068	EF0P000
EF06079	OURSYS*	OURDMP*	EF06070	EF06071	EF06072	EF06073
	EF06074	EF06075	EF06076	EF06077	EF06078	EF0P000
EF06101	OURSYS*	OURDMP*	EF06100	EF0P000		
EF06109	OURSYS*	OURDMP*	EF06108	EF0P000		
EF06110	OURSYS*	OURDMP*	EF06108	EF0P000		
EF06111	OURSYS*	OURDMP*	EF06108	EF0P000		
EF06122	OURSYS*	OURDMP*	EF06112	EF06113	EF06114	EF06115
	EF06116	EF06117	EF06118	EF06119	EF0P000	
EF06140	OURSYS*	OURDMP*	EF06132	EF06133	EF06134	EF06135
	EF06136	EF06137	EF06138	EF06139	EF0P000	
EF06150	OURSYS*	OURDMP*	EF06142	EF06143	EF06144	EF06145
	EF06146	EF06147	EF06148	EF06149	EF0P000	
EF06160	OURSYS*	OURDMP*	EF06152	EF06153	EF06154	EF06155
	EF06156	EF06157	EF06158	EF06159	EF0P000	
EF06170	OURSYS*	OURDMP*	EF06162	EF06163	EF06164	EF06165
	EF06166	EF06167	EF06168	EF06169	EF0P000	
EF06180	OURSYS*	OURDMP*	EF06172	EF06173	EF06174	EF06175
	EF06176	EF06177	EF06178	EF06179	EF0P000	
EF06190	OURSYS*	OURDMP*	EF06182	EF06183	EF06184	EF06185
	EF06186	EF06187	EF06188	EF06189	EF0P000	
EF06199	OURSPC*	OURSYS*	OURDMP*	EF06191	EF06192	EF06193
	EF06194	EF06195	EF06196	EF06197	EF06198	EF0P000
EF0619I	OURSPC*	OURSYS*	OURDMP*	EF0619A	EF0619B	EF0619C
	EF0619D	EF0619E	EF0619F	EF0619G	EF0619H	EF0P000
EF0619R	OURSPC*	OURSYS*	OURDMP*	EF0619J	EF0619K	EF0619L
	EF0619M	EF0619N	EF0619D	EF0619P	EF0619Q	EF0P000
EFD6201	OURSYS*	OURDMP*	EF0P000			
EFD6209	OURSYS*	OURDMP*	EF0P000			
EFD6210	OURSYS*	OURDMP*	EF0P000			
EFD6211	OURSYS*	OURDMP*	EF0P000			
EFD6222	OURSYS*	OURDMP*	EF0P000	EFD6213	EFD6214	EFD6215
	EFD6216	EFD6217	EFD6218	EFD6219		
EFD6233	OURSYS*	OURDMP*	EF0P000	EFD6223	EFD6224	EFD6225
	EFD6226	EFD6227	EFD6228	EFD6229	EFD6230	EFD6231
	EFD6232					
EFD6253	OURSYS*	OURDMP*	EF0P000	EFD6243	EFD6244	EFD6245
	EFD6246	EFD6247	EFD6248	EFD6249	EFD6250	EFD6251

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)					
EFD6269	EFD6252	OURSYSX	OURDMPX	EF0P000	EFD6260	EFD6261
EFD6279	EFD6263	OURSYSX	OURDMPX	EF0P000	EFD6266	EFD6267
EFD6301	EFD6273	OURSYSX	OURDMPX	EF0P000	EFD6270	EFD6271
EFD6309	EFD6274	OURSYSX	OURDMPX	EF0P000	EFD6276	EFD6277
EFD6310	EFD6275	OURSYSX	OURDMPX	EF0P000	EFD6300	EFD6262
EFD6311	EFD6276	OURSYSX	OURDMPX	EF0P000	EFD6308	EFD6268
EFD6322	EFD6301	OURSYSX	OURDMPX	EF0P000	EFD6308	EFD6272
EFD6340	EFD6310	OURSYSX	OURDMPX	EF0P000	EFD6308	EFD6278
EFD6350	EFD6311	OURSYSX	OURDMPX	EF0P000	EFD6312	EFD6313
EFD6360	EFD6315	OURSYSX	OURDMPX	EF0P000	EFD6318	EFD6314
EFD6370	EFD6316	OURSYSX	OURDMPX	EF0P000	EFD6332	EFD6333
EFD6380	EFD6335	OURSYSX	OURDMPX	EF0P000	EFD6338	EFD6334
EFD6390	EFD6336	OURSYSX	OURDMPX	EF0P000	EFD6342	EFD6339
EF0642B	EFD6345	OURSYSX	OURDMPX	EF0P000	EFD6348	EFD6344
EF06803	EFD6346	OURSYSX	OURDMPX	EF0P000	EFD6352	EFD6349
EF06807	EFD6355	OURSYSX	OURDMPX	EF0P000	EFD6358	EFD6353
EF06809	EFD6365	OURSYSX	OURDMPX	EF0P000	EFD6362	EFD6359
EF0680B	EFD6366	OURSYSX	OURDMPX	EF0P000	EFD6368	EFD6363
EF0680D	EFD6375	OURSYSX	OURDMPX	EF0P000	EFD6372	EFD6369
EF0680F	EFD6376	OURSYSX	OURDMPX	EF0P000	EFD6378	EFD6373
EFD680H	EFD6385	OURSYSX	OURDMPX	EF0P000	EFD6382	EFD6379
EFD680J	EFD6386	OURSYSX	OURDMPX	EF0P000	EFD6388	EFD6384
EF06811	OURSPCX	OURSYSX	OURDMPX	EF06422	EF06423	EFD6385
EF06815	EF06425	OURSYSX	OURDMPX	EF06427	EF06428	EF06424
EF06817	EF0P000	OURSYSX	OURDMPX	EF06428	EF06429	EF0642A
EF06819	OURSYSX	OURDMPX	OURDMPX	EF06802	EF0P000	EF06803
EF06821	OURSYSX	OURDMPX	OURDMPX	EF06806	EF0P000	EF06807
EF06823	OURSPCX	OURSYSX	OURDMPX	EF06808	EF0P000	EF06809
EF06825	OURSPCX	OURSYSX	OURDMPX	EF0680A	EF0P000	EF0680B
EFN9301	OURSPCX	OURSYSX	OURDMPX	EF0680C	EF0P000	EF0680D
EFN9302	OURSPCX	OURSYSX	OURDMPX	EF0680E	EF0P000	EF0680F
EFN9303	OURSPCX	OURSYSX	OURDMPX	EF0P000	EF0680G	EF0680H
EF09501	OURSPCX	OURSYSX	OURDMPX	EF0P000	EF0680I	EF0680J
EF09502	OURSPCX	OURSYSX	OURDMPX	EF06810	EF0P000	EF06811
EF09503	OURSPCX	OURSYSX	OURDMPX	EF06814	EF0P000	EF06815
EF09504	OURSPCX	OURSYSX	OURDMPX	EF06816	EF0P000	EF06817
EF09505	OURSPCX	OURSYSX	OURDMPX	EF06818	EF0P000	EF06819
EF09506	OURSPCX	OURSYSX	OURDMPX	EF06820	EF0P000	EF06821
EF09507	OURSPCX	OURSYSX	OURDMPX	EF06822	EF0P000	EF06823
EF09508	OURSPCX	OURSYSX	OURDMPX	EF06824	EF0P000	EF06825
EF09509	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EFN9301
EFN9511	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EFN9302
EF09600	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EFN9303
EF09601	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09501
EF09602	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09502
EF09603	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09503
EF09604	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09504
EF09605	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09505
EF09606	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09506
EF09607	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09507
EFN9611	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09508
EFN9612	OURSPCX	OURSYSX	OURDMPX	EF09500	EF0P000	EF09509
EFN9613	OURSPCX	OURSYSX	OURDMPX	EFN9610	EFN9610	EFN9511

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
EF09711	OURSYS*	EF09710	EF0P000			
EF09713	OURSYS*	EF09712	EF0P000			
EF09720	OURSPC*	OURSYS*	OURDMP*	EF09500	EF0P000	
EF09721	OURSPC*	OURSYS*	OURDMP*	EF09500	EF0P000	
EF09731	OURSPC*	OURSYS*	OURDMP*	EF09500	EF0P000	
EF09901	OURSYS*	EF09900	EF0P000			
EF09902	OURSYS*	EF09900	EF0P000			
EF09903	OURSYS*	EF09900	EF0P000			
EFM9A01	OURSPC*	OURSYS*	EFM9A00			
EFM9A02	OURSPC*	OURSYS*	EFM9A00			
EFM9A03	OURSPC*	OURSYS*	EFM9A00			
EF09B01	OURSPC*	OURSYS*	OURDMP*	EF09500	EF0P000	
EFD9C00	OURSPC*	OURSYS*	EF0P000	EFD9000		
EFD9C01	OURSPC*	OURSYS*	EF0P000	EFD9000	EFD9200	
EFD9C02	OURSPC*	OURSYS*	EF0P000	EFD9000	EFD9200	
EFD9C03**	OURSPC*	OURSYS*	EF0P000	EFD9000	EFD9200	
EFD9C04**	OURSPC*	OURSYS*	EF0P000	EFD9000	EFD9200	
EF0C301	OURSYS*	OURDMP*	EF0C100	EF0C101	EF0C300	EF0P000
EF0C302	OURSYS*	OURDMP*	EF0C100	EF0C101	EF0C300	EF0P000
EF0C304	OURSYS*	OURDMP*	EF0C102	EF0C103	EF0C303	EF0P000
EF0C305	OURSYS*	OURDMP*	EF0C102	EF0P000		
EF0C306	OURSYS*	OURDMP*	EF0C102	EF0P000		
EF0C307	OURSYS*	OURDMP*	EF0C102	EF0P000		
EFDC311	OURSYS*	OURDMP*	EF0P000	EFDC104	EFDC105	EFDC310
EFDC314	OURSYS*	OURDMP*	EF0P000	EFDC106	EFDC107**	EFDC313
EFDD600	OURSPC*	OURSYS*	EF0P000			
EFDD601	OURSPC*	OURSYS*	EF0P000			
EFDD602	OURSPC*	OURSYS*	EF0P000			
EFDD603	OURSPC*	OURSYS*	EF0P000			
EFDD604	OURSPC*	OURSYS*	EF0P000			
EFDD605	OURSPC*	OURSYS*	EF0P000			
EFDD606	OURSPC*	OURSYS*	EF0P000			
EFDD607	OURSPC*	OURSYS*	EF0P000			
EFDD608*	OURSPC*	OURSYS*	EF0P000			
EFDD609	OURSPC*	OURSYS*	EF0P000			
EFDD610	OURSPC*	OURSYS*	EF0P000			
EFDD611	OURSPC*	OURSYS*	EF0P000			
EFDDA01	OURSPC*	OURSYS*	EF0P000			
EFDDA02	OURSPC*	OURSYS*	EF0P000			
EFDE220	OURSYS*	OURDMP*	EF0P000	EFDE000	EFDE201	EFDE202
	EFDE203	EFDE204	EFDE205	EFDE206	EFDE207	EFDE208
	EFDE209	EFDE20A	EFDE211	EFDE212	EFDE213	EFDE214
	EFDE215	EFDE216				
EFDE250	OURSYS*	OURDMP*	EF0P000	EFDE000	EFDE231	EFDE232
	EFDE233	EFDE234	EFDE235	EFDE236	EFDE237	EFDE238
	EFDE239	EFDE23A	EFDE241	EFDE242	EFDE243	EFDE244
	EFDE245	EFDE246				
EFDE420	OURSYS*	OURDMP*	EF0P000	EFDE000	EFDE401	EFDE402
	EFDE403	EFDE404	EFDE405	EFDE406	EFDE407	EFDE408
	EFDE409	EFDE40A	EFDE411	EFDE412	EFDE413	EFDE414
	EFDE415	EFDE416				
EFDE450	OURSYS*	OURDMP*	EF0P000	EFDE000	EFDE431	EFDE432
	EFDE433	EFDE434	EFDE435	EFDE436	EFDE437	EFDE438
	EFDE439	EFDE43A	EFDE441	EFDE442	EFDE443	EFDE444
	EFDE445	EFDE446				
EFDF000	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EFDF001	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EFDF002	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EFDF003	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EFDF004	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EFDF005	OURSYS*	OURDMP*	MATHFUN*	EF0P000		
EGD0001	OURSYS*	MATHFUN*	EG0P000	EGD0000		
EGD0003	OURSYS*	EG0P000	EGD0002			

** Multiple versions of source file supplied(USE and ADA file types)

** Source file supplied with DEC file type only

* Source file supplied with USE file type only

Table 7-3 Ada Type E Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
EGD0009	OURSYS*	EGOP000	EGD0004	EGD0005	EGD0006	EGD0007
EG0000C	OURSYS*	OURDMP*	EGOP000			
EGD0000E	OURSYS*	MATHFUN*	EGOP000	EGD0000	EGD0000D	
EGN0000K	OURSYS*	EGN000F	EGN000G	EGN000H	EGN000I	EGN000J
EGD00039	OURSPC*	OURSYS*	EGOP000	EGD0004	EGD0005	EGD0006
	EGD0007	EGD0008	EGD0019	EGD0029		
EGD00049	OURSPC*	OURSYS*	EGOP000	EGD0004	EGD0005	EGD0006
	EGD0007	EGD0008	EGD0019	EGD0029		
EGD00059**	OURSPC*	OURSYS*	EGOP000	EGD0004	EGD0005	EGD0006
	EGD0007	EGD0008	EGD0019	EGD0029		
EL09101	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	
EL09111	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	
EL09121	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	
EL09131	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	
ELD9202	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9203	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9204**	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9212	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9213	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9214**	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9222	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9223	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9224**	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9232	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9233	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELD9234**	OURSPC*	OURSYS*	EL09000	EL09100	EL0P000	ELD9200
ELDE2C1	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B1	
ELDE2C2	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B1	
ELDE2C3**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B1	
ELDE2C4**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B1	
ELDE2C5	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B2	
ELDE2C6	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B2	
ELDE2C7**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B2	
ELDE2C8**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE2B2	
ELDE4C1	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B1	
ELDE4C2	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B1	
ELDE4C3**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B1	
ELDE4C4**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B1	
ELDE4C5	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B2	
ELDE4C6	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B2	
ELDE4C7**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B2	
ELDE4C8**	OURSPC*	OURSYS*	EL0P000	ELDE000	ELDE4B2	
E000104	OURSYS*	OURDMP*	E000002	E000102	E000103	E00P000
E000109	OURSYS*	OURDMP*	E000001	E000107	E000108	E00P000
E000114	OURSYS*	OURDMP*	E000001	E000112	E000113	E00P000
E000204	OURSYS*	OURDMP*	E000202	E000203	E00P000	
E000209	OURSYS*	OURDMP*	E000001	E000207	E000208	E00P000
E000300	OURSYS*	E00P000				
E000305	OURSYS*	E00P000				
E000310	OURSYS*	E00P000				
E000313	OURSYS*	OURDMP*	E000311	E000312	E00P000	
E000316	OURSYS*	OURDMP*	E000314	E000315	E00P000	
E000319	OURSYS*	OURDMP*	E000317	E000318	E00P000	
E000322	OURSYS*	OURDMP*	E000320	E000321	E00P000	
E000325	OURSYS*	OURDMP*	E000323	E000324	E00P000	
E000328	OURSYS*	OURDMP*	E000326	E000327	E00P000	
E000331	OURSYS*	OURDMP*	E000329	E000330	E00P000	
E000334	OURSYS*	OURDMP*	E000332	E000333	E00P000	
E000337	OURSYS*	OURDMP*	E000335	E000336	E00P000	
E000340	OURSYS*	OURDMP*	E000338	E000339	E00P000	
E000343	OURSYS*	OURDMP*	E000341	E000342	E00P000	
E000346	OURSYS*	OURDMP*	MATHFUN*	E00P000	E000344	E000345
E000350	OURSYS*	OURDMP*	E000347	E000348	E000349	E00P000

** Source file supplied with DEC file type only

Table 7-3 Ada Type E Test Programs and Source Code Files (concluded)

PROGRAM	SUPPORT FILES(X - TEST SUPPORT SOFTWARE PACKAGES USED)					
E000353	OURSYSX	OURDMPX	E000347	E000351	E000352	E00P000
E000356	OURSYSX	OURDMPX	E000354	E000355	E00P000	
E000359	OURSYSX	OURDMPX	E000357	E000358	E00P000	
E000362	OURSYSX	OURDMPX	E000360	E000361	E00P000	
E000365	OURSPCX	OURSYSX	OURDMPX	E000363	E000364	E00P000
E000368	OURSPCX	OURSYSX	OURDMPX	E000366	E000367	E00P000
E00036B	OURSPCX	OURSYSX	OURDMPX	E000369	E00036A	E00P000
E0D036E	OURSPCX	OURSYSX	OURDMPX	E00P000	E0D036C	E0D036D
E00036H	OURSPCX	OURSYSX	OURDMPX	E00036F	E00036G	E00P000
E00036K	OURSPCX	OURSYSX	OURDMPX	E00036I	E00036J	E00P000
E00036N	OURSPCX	OURSYSX	OURDMPX	E00036L	E00036M	E00P000
E0D036Q	OURSPCX	OURSYSX	OURDMPX	E00P000	E0D036O	E0D036P
E0D036T	OURSPCX	OURSYSX	OURDMPX	E00P000	E0D036R	E0D036S
E00036W	OURSPCX	OURSYSX	OURDMPX	E00036U	E00036V	E00P000
E0D036Z	OURSYSX	OURDMPX	OURTYPX	E00P000	E0D036X	E0D036Y
E0D0373	OURSPCX	OURSYSX	OURDMPX	E00P000	E0D0370	E0D0371
	E0D0372					
E000378	OURSYSX	OURDMPX	E000374	E000375	E000376	E000377
	E00P000					
E0D0383	OURSPCX	OURSYSX	OURDMPX	E000374	E00P000	E0D0380
	E0D0381	E0D0382				
E000391	OURSYSX	OURDMPX	E000390	E00P000		
E000404	OURSYSX	OURDMPX	E000402	E000403	E00P000	
E000500	OURSYSX	E00P000				
E000504	OURSYSX	OURDMPX	E000502	E000503	E00P000	
E000509	OURSYSX	OURDMPX	E000507	E000508	E00P000	
E000514	OURSYSX	OURDMPX	E000512	E000513	E00P000	
E000519	OURSYSX	OURDMPX	E000517	E000518	E00P000	
E0D0525	OURSYSX	OURDMPX	E00P000	E0D0523	E0D0524	
E0D0529	OURSYSX	OURDMPX	E00P000	E0D0526	E0D0527	E0D0528
E000604	OURSYSX	OURDMPX	E000602	E000603	E00P000	
E000605	OURSYSX	OURDMPX	E00P000			
E000606	OURSYSX	OURDMPX	E00P000			
E000607	OURSYSX	OURDMPX	E00P000			
E000704	OURSYSX	OURDMPX	E000702	E000703	E00P000	
E000705	OURSYSX	OURDMPX	E00P000			
E000708	OURSYSX	OURDMPX	E000706	E000707	E00P000	
E000711	OURSYSX	OURDMPX	E000709	E000710	E00P000	
E000713	OURSYSX	OURDMPX	E00P000			
E000714	OURSYSX	OURDMPX	E00P000			
E000715	OURSYSX	OURDMPX	E00P000			
E000716	OURSYSX	OURDMPX	E00P000			
E000717	OURSYSX	OURDMPX	E00P000			
E000718	OURSYSX	OURDMPX	E00P000			
E000721	OURSYSX	OURDMPX	E000719	E000720	E00P000	

Table G-4. Ada Type S Test Programs and Source Code Files

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SA00000	OURSYS*	OURDMP*	SA0P000			
SF03519	OURSYS*	OURDMP*	SF03500	SF03501	SF03502	SF03503
	SF03504	SF03509	SF03510	SF03511	SF03512	SF03513
	SF03514	SF03517	SF03518	SF0P000		
SF03550	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD3551	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF03620	OURSYS*	OURDMP*	SF03600	SF03601	SF03602	SF03603
	SF03604	SF03605	SF03606	SF03607	SF03608	SF03609
	SF03610	SF03611	SF03612	SF03613	SF03614	SF03615
	SF03616	SF03617	SF03618	SF03619	SF0P000	
SF03650	OURSYS*	OURDMP*	SF03630	SF03631	SF03632	SF03633
	SF03634	SF03635	SF03636	SF03637	SF03638	SF03639
	SF03641	SF03642	SF03643	SF03644	SF03645	SF03646
	SF03647	SF03648	SF03649	SF0P000		
SF03704	OURSPC*	OURSYS*	OURDMP*	SF03700	SF03701	SF03702
	SF03703	SF0P000				
SF03805	OURSPC*	OURSYS*	OURDMP*	SF03800	SF03801	SF03802
	SF03803	SF03804	SF0P000			
SF04120	OURSYS*	OURDMP*	SF0P000			
SF04121	OURSYS*	OURDMP*	SF0P000			
SF04122	OURSYS*	OURDMP*	SF0P000			
SF04123	OURSYS*	OURDMP*	SF0P000			
SF04124	OURSYS*	OURDMP*	SF0P000			
SF04125	OURSYS*	OURDMP*	SF0P000			
SF04126	OURSYS*	OURDMP*	SF0P000			
SF04127	OURSYS*	OURDMP*	SF0P000			
SFD4128*	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04129	OURSYS*	OURDMP*	SF0P000			
SFD412A	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0412B	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD412C	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD412D	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04130	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04131	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04132	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04133	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD4135	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD4136	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD4137	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD4138	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD4139	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD413A	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD413B	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD413C	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0413D	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD413E	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD413G	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0413H	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0413I	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0413J	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0413K	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04310	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04311	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04312	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04510	OURSYS*	OURDMP*	SF0P000			
SF04511	OURSYS*	OURDMP*	SF0P000			
SF04512	OURSYS*	OURDMP*	SF0P000			
SF04513	OURSYS*	OURDMP*	SF0P000			
SF04514	OURSYS*	OURDMP*	SF0P000			
SF04515	OURSYS*	OURDMP*	SF0P000			
SF04516	OURSYS*	OURDMP*	SF0P000			
SF04517	OURSYS*	OURDMP*	SF0P000			
SF0451J	OURSYS*	OURDMP*	SF04518	SF04519	SF0451A	SF0451B
	SF0451C	SF0451D	SF0451E	SF0451F	SF0451G	SF0451H
	SF0451I	SF0P000				

* Source file supplied with USE file type only

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SFD451U*	OURSYS*	OURDMP*	OURTYP*	SF0P000	SFD451K	SFD451L
	SFD451M	SFD451N	SFD451O	SFD451P	SFD451Q	SFD451R
	SFD451S					
SF0451Z	OURSYS*	OURDMP*	SF0451V	SF0451W	SF0451X	SF0451Y
	SF0P000					
SF04520	OURSYS*	OURDMP*	SF0P000			
SF04521	OURSYS*	OURDMP*	SF0P000			
SF04522	OURSYS*	OURDMP*	SF0P000			
SFD4523	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4524	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4525	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4526	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SF04527	OURSYS*	OURDMP*	SF0P000			
SFD4528	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04529	OURSYS*	OURDMP*	SF0P000			
SFD452A	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF04530	OURSYS*	OURDMP*	SF0P000			
SF04531	OURSYS*	OURDMP*	SF0P000			
SF04532	OURSYS*	OURDMP*	SF0P000			
SF04533	OURSYS*	OURDMP*	SF0P000			
SF04534	OURSYS*	OURDMP*	SF0P000			
SF04535	OURSYS*	OURDMP*	SF0P000			
SF04536	OURSYS*	OURDMP*	SF0P000			
SF04537	OURSYS*	OURDMP*	SF0P000			
SF04538	OURSYS*	OURDMP*	SF0P000			
SF04539	OURSYS*	OURDMP*	SF0P000			
SF0453A	OURSYS*	OURDMP*	SF0P000			
SF0453B	OURSYS*	OURDMP*	SF0P000			
SF0453C	OURSYS*	OURDMP*	SF0P000			
SF04540	OURSYS*	OURDMP*	SF0P000			
SF04541	OURSYS*	OURDMP*	SF0P000			
SF04550	OURSYS*	OURDMP*	SF0P000			
SF04551	OURSYS*	OURDMP*	SF0P000			
SF04552	OURSYS*	OURDMP*	SF0P000			
SF04553	OURSYS*	OURDMP*	SF0P000			
SF04554	OURSYS*	OURDMP*	SF0P000			
SF04555	OURSYS*	OURDMP*	SF0P000			
SF04556	OURSYS*	OURDMP*	SF0P000			
SF04557	OURSYS*	OURDMP*	SF0P000			
SF04558	OURSYS*	OURDMP*	SF0P000			
SF04559	OURSYS*	OURDMP*	SF0P000			
SFD455D	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455E	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455F	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455G	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455I	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455J	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455K	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD455L	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SF0455N	OURSYS*	OURDMP*	SF0P000			
SF0455O	OURSYS*	OURDMP*	SF0P000			
SF04560	OURSYS*	OURDMP*	SF0P000			
SF04562	OURSYS*	OURDMP*	SF0P000			
SF04563	OURSYS*	OURDMP*	SF0P000			
SFD4566	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4567	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFL4568	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4569	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SF04600	OURSYS*	OURDMP*	SF0P000			
SF04601	OURSYS*	OURDMP*	SF0P000			
SF04602	OURSYS*	OURDMP*	SF0P000			
SFD4603	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4604	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4605	OURSYS*	OURDMP*	OURTYP*	SF0P000		
SFD4606	OURSYS*	OURDMP*	OURTYP*	SF0P000		

* Source file supplied with USE file type only

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>				
SFD4607	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD4608	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD4609	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD460A	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD460B	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SF0460E	OURSYS*	OURDMP*	SFOP000		
SF0460F	OURSYS*	OURDMP*	SFOP000		
SF0460G	OURSYS*	OURDMP*	SFOP000		
SF0460H	OURSYS*	OURDMP*	SFOP000		
SF0460I	OURSYS*	OURDMP*	SFOP000		
SF0460J	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD460K	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD460L	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SF0460M	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD460N	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD460O	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD4804	OURSPC*	OURSYS*	OURDMP*	SFOP000	SFD4800 SFD4801
	SFD4802	SFD4803**			
SF05200	OURSYS*	OURDMP*	SFOP000		
SF05201	OURSYS*	OURDMP*	SFOP000		
SF05202	OURSYS*	OURDMP*	SFOP000		
SF05203	OURSYS*	OURDMP*	SFOP000		
SF05204	OURSYS*	OURDMP*	SFOP000		
SF05205	OURSYS*	OURDMP*	SFOP000		
SF05206	OURSYS*	OURDMP*	SFOP000		
SF05207	OURSYS*	OURDMP*	SFOP000		
SF05208	OURSYS*	OURDMP*	SFOP000		
SF05209	OURSYS*	OURDMP*	SFOP000		
SF0520A	OURSYS*	OURDMP*	SFOP000		
SF0520B	OURSYS*	OURDMP*	SFOP000		
SFD520C	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520D	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520E	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520F	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520G	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520H	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520I	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD520J	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SF0520M	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD520N	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SF0520O	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD520P	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SFD520Q	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SF0520R	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SF0520S	OURSPC*	OURSYS*	OURDMP*	SFOP000	
SF05210	OURSYS*	OURDMP*	SFOP000		
SF05211	OURSYS*	OURDMP*	SFOP000		
SF05212	OURSYS*	OURDMP*	SFOP000		
SF05213	OURSYS*	OURDMP*	SFOP000		
SF05214	OURSYS*	OURDMP*	SFOP000		
SF05215	OURSYS*	OURDMP*	SFOP000		
SFD5218	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD5219	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521A	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521B	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521C	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521D	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521E	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SFD521F	OURSYS*	OURDMP*	OURTYP*	SFOP000	
SF0521H	OURSYS*	OURDMP*	SFOP000		
SF0521I	OURSYS*	OURDMP*	SFOP000		
SF0521J	OURSYS*	OURDMP*	SFOP000		
SF0521K	OURSYS*	OURDMP*	SFOP000		
SFD521K	OURSPC*	OURSYS*	OURDMP*	SFOP000	

** Multiple versions of source file supplied(USE and ADA file types)

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SF0521L	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD521M	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF0521N	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD521O	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SFD521P	OURSPC*	OURSYS*	OURDMP*	SF0P000		
SF05304	OURSYS*	OURDMP*	SF05300	SF05301	SF05302	SF05303
	SF0P000					
SF05308	OURSYS*	OURDMP*	SF05305	SF05306	SF05307	SF0P000
SF05408	OURSYS*	OURDMP*	SF05400	SF05401	SF05402	SF05403
	SF05404	SF05405	SF05406	SF05407	SF0P000	
SF05505	OURSYS*	OURDMP*	SF05501	SF05502	SF05503	SF05504
	SF0P000					
SF0550C	OURSYS*	OURDMP*	SF05506	SF05507	SF05508	SF05509
	SF0550A	SF0550B	SF0P000			
SF0550D	OURSYS*	OURDMP*	SF0P000			
SF0550E	OURSYS*	OURDMP*	SF0P000			
SF0550F	OURSYS*	OURDMP*	SF0P000			
SF06001	OURSYS*	OURDMP*	SF0P000			
SF06009	OURSYS*	OURDMP*	SF0P000			
SF06010	OURSYS*	OURDMP*	SF0P000			
SF06011	OURSYS*	OURDMP*	SF0P000			
SF06022	OURSYS*	OURDMP*	SF06013	SF06014	SF06015	SF06016
	SF06017	SF06018	SF06019	SF0P000		
SF06033	OURSYS*	OURDMP*	SF06023	SF06024	SF06025	SF06026
	SF06027	SF06028	SF06029	SF06030	SF06031	SF06032
	SF0P000					
SF06053	OURSYS*	OURDMP*	SF06043	SF06044	SF06045	SF06046
	SF06047	SF06048	SF06049	SF06050	SF06051	SF06052
	SF0P000					
SF06069	OURSYS*	OURDMP*	SF06060	SF06061	SF06062	SF06063
	SF06064	SF06065	SF06066	SF06067	SF06068	SF0P000
SF06079	OURSYS*	OURDMP*	SF06070	SF06071	SF06072	SF06073
	SF06074	SF06075	SF06076	SF06077	SF06078	SF0P000
SF06101	OURSYS*	OURDMP*	SF06100	SF0P000		
SF06109	OURSYS*	OURDMP*	SF06108	SF0P000		
SF06110	OURSYS*	OURDMP*	SF06108	SF0P000		
SF06111	OURSYS*	OURDMP*	SF06108	SF0P000		
SF06122	OURSYS*	OURDMP*	SF06112	SF06113	SF06114	SF06115
	SF06116	SF06117	SF06118	SF06119	SF0P000	
SF06140	OURSYS*	OURDMP*	SF06132	SF06133	SF06134	SF06135
	SF06136	SF06137	SF06138	SF06139	SF0P000	
SF06150	OURSYS*	OURDMP*	SF06142	SF06143	SF06144	SF06145
	SF06146	SF06147	SF06148	SF06149	SF0P000	
SF06160	OURSYS*	OURDMP*	SF06152	SF06153	SF06154	SF06155
	SF06156	SF06157	SF06158	SF06159	SF0P000	
SF06170	OURSYS*	OURDMP*	SF06162	SF06163	SF06164	SF06165
	SF06166	SF06167	SF06168	SF06169	SF0P000	
SF06180	OURSYS*	OURDMP*	SF06172	SF06173	SF06174	SF06175
	SF06176	SF06177	SF06178	SF06179	SF0P000	
SF06190	OURSYS*	OURDMP*	SF06182	SF06183	SF06184	SF06185
	SF06186	SF06187	SF06188	SF06189	SF0P000	
SF06199	OURSPC*	OURSYS*	OURDMP*	SF06191	SF06192	SF06193
	SF06194	SF06195	SF06196	SF06197	SF06198	SF0P000
SF0619I	OURSPC*	OURSYS*	OURDMP*	SF0619A	SF0619B	SF0619C
	SF0619D	SF0619E	SF0619F	SF0619G	SF0619H	SF0P000
SF0619R	OURSPC*	OURSYS*	OURDMP*	SF0619J	SF0619K	SF0619L
	SF0619M	SF0619N	SF0619O	SF0619P	SF0619Q	SF0P000
SFD6201	OURSYS*	OURDMP*	SF0P000			
SFD6209	OURSYS*	OURDMP*	SF0P000			
SFD6210	OURSYS*	OURDMP*	SF0P000			
SFD6211	OURSYS*	OURDMP*	SF0P000			
SFD6222	OURSYS*	OURDMP*	SF0P000	SFD6213	SFD6214	SFD6215
	SFD6216	SFD6217	SFD6218	SFD6219		
SFD6233	OURSYS*	OURDMP*	SF0P000	SFD6223	SFD6224	SFD6225
	SFD6226	SFD6227	SFD6228	SFD6229	SFD6230	SFD6231

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SFD6253	SFD6232 OURSYS*	OURDMP*	SF0P000	SFD6243	SFD6244	SFD6245
	SFD6246	SFD6247	SFD6248	SFD6249	SFD6250	SFD6251
	SFD6252					
SFD6269	OURSYS*	OURDMP*	SF0P000	SFD6260	SFD6261	SFD6262
	SFD6263	SFD6264	SFD6265	SFD6266	SFD6267	SFD6268
SFD6279	OURSYS*	OURDMP*	SF0P000	SFD6270	SFD6271	SFD6272
	SFD6273	SFD6274	SFD6275	SFD6276	SFD6277	SFD6278
SFD6301	OURSYS*	OURDMP*	SF0P000	SFD6300		
SFD6309	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6310	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6311	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6322	OURSYS*	OURDMP*	SF0P000	SFD6312	SFD6313	SFD6314
	SFD6315	SFD6316	SFD6317	SFD6318	SFD6319	
SFD6340	OURSYS*	OURDMP*	SF0P000	SFD6332	SFD6333	SFD6334
	SFD6335	SFD6336	SFD6337	SFD6338	SFD6339	
SFD6350	OURSYS*	OURDMP*	SF0P000	SFD6342	SFD6343	SFD6344
	SFD6345	SFD6346	SFD6347	SFD6348	SFD6349	
SFD6360	OURSYS*	OURDMP*	SF0P000	SFD6352	SFD6353	SFD6354
	SFD6355	SFD6356	SFD6357	SFD6358	SFD6359	
SFD6370	OURSYS*	OURDMP*	SF0P000	SFD6362	SFD6363	SFD6364
	SFD6365	SFD6366	SFD6367	SFD6368	SFD6369	
SFD6380	OURSYS*	OURDMP*	SF0P000	SFD6372	SFD6373	SFD6374
	SFD6375	SFD6376	SFD6377	SFD6378	SFD6379	
SFD6390	OURSYS*	OURDMP*	SF0P000	SFD6382	SFD6383	SFD6384
	SFD6385	SFD6386	SFD6387	SFD6388	SFD6389	
SF0642B	OURSPC*	OURDMP*	OURDMP*	SF06422	SF06423	SF06424
	SF06425	SF06426	SF06427	SF06428	SF06429	SF0642A
	SF0P000					
SF06803	OURSYS*	OURDMP*	SF06802	SF0P000		
SF06807	OURSYS*	OURDMP*	SF06806	SF0P000		
SF06809	OURSPC*	OURSYS*	OURDMP*	SF06808	SF0P000	
SF0680B	OURSPC*	OURSYS*	OURDMP*	SF0680A	SF0P000	
SF0680D	OURSPC*	OURSYS*	OURDMP*	SF0680C	SF0P000	
SF0680F	OURSPC*	OURSYS*	OURDMP*	SF0680E	SF0P000	
SFD680H	OURSPC*	OURSYS*	OURDMP*	SF0P000	SFD680G	
SFD680J	OURSPC*	OURSYS*	OURDMP*	SF0P000	SFD680I	
SF06811	OURSYS*	OURDMP*	SF06810	SF0P000		
SF06815	OURSYS*	OURDMP*	SF06814	SF0P000		
SF06817	OURSYS*	OURDMP*	SF06816	SF0P000		
SF06819	OURSYS*	OURDMP*	SF06818	SF0P000		
SF06821	OURSYS*	OURDMP*	SF06820	SF0P000		
SF06823	OURSPC*	OURSYS*	OURDMP*	SF06822	SF0P000	
SF06825	OURSPC*	OURSYS*	OURDMP*	SF06824	SF0P000	
SFN9301	OURSPC*	OURSYS*	SFN9300			
SFN9302	OURSPC*	OURSYS*	SFN9300			
SFN9303	OURSPC*	OURSYS*	SFN9300			
SF09501	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09502	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09503	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09504	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09505	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09506	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09507	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09508	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09509	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SFN9511	OURSPC*	OURSYS*	SFN9510			
SF09600	OURSPC*	OURSYS*	SF0P000			
SF09601	OURSPC*	OURSYS*	SF0P000			
SF09602	OURSPC*	OURSYS*	SF0P000			
SF09603	OURSPC*	OURSYS*	SF0P000			
SF09604	OURSPC*	OURSYS*	SF0P000			
SF09605	OURSPC*	OURSYS*	SF0P000			
SF09606	OURSPC*	OURSYS*	SF0P000			
SF09607	OURSPC*	OURSYS*	SF0P000			

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SFN9611	OURSPC*	OURSYS*				
SFN9612	OURSPC*	OURSYS*	SFN9610			
SFN9613	OURSPC*	OURSYS*	SFN9610			
SF09711	OURSYS*	SF09710	SF0P000			
SF09713	OURSYS*	SF09712	SF0P000			
SF09720	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09721	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09731	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09901	OURSYS*	SF09900	SF0P000			
SF09902	OURSYS*	SF09900	SF0P000			
SF09903	OURSYS*	SF09900	SF0P000			
SFM9A01	OURSPC*	OURSYS*	SFM9A00			
SFM9A02	OURSPC*	OURSYS*	SFM9A00			
SFM9A03	OURSPC*	OURSYS*	SFM9A00			
SF09B01	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SFD9C00	OURSPC*	OURSYS*	SF0P000	SFD9000		
SFD9C01	OURSPC*	OURSYS*	SF0P000	SFD9000	SFD9200	
SFD9C02	OURSPC*	OURSYS*	SF0P000	SFD9000	SFD9200	
SFD9C03**	OURSPC*	OURSYS*	SF0P000	SFD9000	SFD9200	
SFD9C04**	OURSPC*	OURSYS*	SF0P000	SFD9000	SFD9200	
SFMB001	OURSYS*	SFMB000				
SFMB003	OURSYS*	SFMB002				
SFMB005	OURSYS*	SFMB004				
SFMB007	OURSYS*	SFMB006				
SFMB009	OURSYS*	SFMB008				
SFMB012	OURSYS*	SFMB010	SFMB011			
SFMB016	OURSYS*	SFMB013	SFMB014	SFMB015		
SFMB020	OURSYS*	SFMB017	SFMB018	SFMB019		
SFMB024	OURSYS*	SFMB021	SFMB022	SFMB023		
SFMB028	OURSYS*	SFMB025	SFMB026	SFMB027		
SFMB032	OURSYS*	SFMB013	SFMB014	SFMB015		
SFMB036	OURSYS*	SFMB033	SFMB034	SFMB035		
SFMB040	OURSYS*	SFMB037	SFMB038	SFMB039		
SFMB044	OURSYS*	SFMB041	SFMB042	SFMB043		
SF0C301	OURSYS*	OURDMP*	SF0C100	SF0C101	SF0C300	SF0P000
SF0C302	OURSYS*	OURDMP*	SF0C100	SF0C101	SF0C300	SF0P000
SF0C304	OURSYS*	OURDMP*	SF0C102	SF0C103	SF0C303	SF0P000
SF0C305	OURSYS*	OURDMP*	SF0C102	SF0P000		
SF0C306	OURSYS*	OURDMP*	SF0C102	SF0P000		
SF0C307	OURSYS*	OURDMP*	SF0C102	SF0P000		
SFDC311	OURSYS*	OURDMP*	SF0P000	SFDC104	SFDC105	SFDC310
SFDC314	OURSYS*	OURDMP*	SF0P000	SFDC106	SFDC107**	SFDC313
SFDD600	OURSPC*	OURSYS*	SF0P000			
SFDD601	OURSPC*	OURSYS*	SF0P000			
SFDD602	OURSPC*	OURSYS*	SF0P000			
SFDD603	OURSPC*	OURSYS*	SF0P000			
SFDD604	OURSPC*	OURSYS*	SF0P000			
SFDD605	OURSPC*	OURSYS*	SF0P000			
SFDD606	OURSPC*	OURSYS*	SF0P000			
SFDD607	OURSPC*	OURSYS*	SF0P000			
SFDD608*	OURSPC*	OURSYS*	SF0P000			
SFDD609	OURSPC*	OURSYS*	SF0P000			
SFDD610	OURSPC*	OURSYS*	SF0P000			
SFDD611	OURSPC*	OURSYS*	SF0P000			
SF0D720	OURSYS*					
SF0D721	OURSYS*					
SFDD722	OURSYS*	OURTYP*				
SFDD723	OURSYS*	OURTYP*				
SFDD724	OURSYS*	OURTYP*				
SFDD725	OURSYS*	OURTYP*				
SF0D727	OURSYS*					
SF0D728	OURSPC*	OURSYS*				
SF0D729	OURSPC*	OURSYS*				

** Multiple versions of source file supplied(USE and ADA file types)

** Source file supplied with DEC file type only

* Source file supplied with USE file type only

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SFD6253	SFD6232 OURSYS* SFD6246 SFD6252	OURDMP* SFD6247	SF0P000 SFD6248	SFD6243 SFD6249	SFD6244 SFD6250	SFD6245 SFD6251
SFD6269	OURSYS* SFD6263	OURDMP* SFD6264	SF0P000 SFD6265	SFD6260 SFD6266	SFD6261 SFD6267	SFD6262 SFD6268
SFD6279	OURSYS* SFD6273	OURDMP* SFD6274	SF0P000 SFD6275	SFD6270 SFD6276	SFD6271 SFD6277	SFD6272 SFD6278
SFD6301	OURSYS*	OURDMP*	SF0P000	SFD6300		
SFD6309	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6310	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6311	OURSYS*	OURDMP*	SF0P000	SFD6308		
SFD6322	OURSYS* SFD6315	OURDMP* SFD6316	SF0P000 SFD6317	SFD6312 SFD6318	SFD6313 SFD6319	SFD6314
SFD6340	OURSYS* SFD6335	OURDMP* SFD6336	SF0P000 SFD6337	SFD6332 SFD6338	SFD6333 SFD6339	SFD6334
SFD6350	OURSYS* SFD6345	OURDMP* SFD6346	SF0P000 SFD6347	SFD6342 SFD6348	SFD6343 SFD6349	SFD6344
SFD6360	OURSYS* SFD6355	OURDMP* SFD6356	SF0P000 SFD6357	SFD6352 SFD6358	SFD6353 SFD6359	SFD6354
SFD6370	OURSYS* SFD6365	OURDMP* SFD6366	SF0P000 SFD6367	SFD6362 SFD6368	SFD6363 SFD6369	SFD6364
SFD6380	OURSYS* SFD6375	OURDMP* SFD6376	SF0P000 SFD6377	SFD6372 SFD6378	SFD6373 SFD6379	SFD6374
SFD6390	OURSYS* SFD6385	OURDMP* SFD6386	SF0P000 SFD6387	SFD6382 SFD6388	SFD6383 SFD6389	SFD6384
SF0642B	OURSPC* SF06425 SF0P000	OURDMP* SF06426	OURDMP* SF06427	SF06422 SF06428	SF06423 SF06429	SF06424 SF0642A
SF06803	OURSYS*	OURDMP*	SF06802	SF0P000		
SF06807	OURSYS*	OURDMP*	SF06806	SF0P000		
SF06809	OURSPC*	OURSYS*	OURDMP*	SF06808	SF0P000	
SF0680B	OURSPC*	OURSYS*	OURDMP*	SF0680A	SF0P000	
SF0680D	OURSPC*	OURSYS*	OURDMP*	SF0680C	SF0P000	
SF0680F	OURSPC*	OURSYS*	OURDMP*	SF0680E	SF0P000	
SFD680H	OURSPC*	OURSYS*	OURDMP*	SF0P000	SFD680G	
SFD680J	OURSPC*	OURSYS*	OURDMP*	SF0P000	SFD680I	
SF06811	OURSYS*	OURDMP*	SF06810	SF0P000		
SF06815	OURSYS*	OURDMP*	SF06814	SF0P000		
SF06817	OURSYS*	OURDMP*	SF06816	SF0P000		
SF06819	OURSYS*	OURDMP*	SF06818	SF0P000		
SF06821	OURSYS*	OURDMP*	SF06820	SF0P000		
SF06823	OURSPC*	OURSYS*	OURDMP*	SF06822	SF0P000	
SF06825	OURSPC*	OURSYS*	OURDMP*	SF06824	SF0P000	
SFN9301	OURSPC*	OURSYS*	SFN9300			
SFN9302	OURSPC*	OURSYS*	SFN9300			
SFN9303	OURSPC*	OURSYS*	SFN9300			
SF09501	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09502	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09503	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09504	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09505	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09506	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09507	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09508	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SF09509	OURSPC*	OURSYS*	OURDMP*	SF09500	SF0P000	
SFN9511	OURSPC*	OURSYS*	SFN9510			
SF09600	OURSPC*	OURSYS*	SF0P000			
SF09601	OURSPC*	OURSYS*	SF0P000			
SF09602	OURSPC*	OURSYS*	SF0P000			
SF09603	OURSPC*	OURSYS*	SF0P000			
SF09604	OURSPC*	OURSYS*	SF0P000			
SF09605	OURSPC*	OURSYS*	SF0P000			
SF09606	OURSPC*	OURSYS*	SF0P000			
SF09607	OURSPC*	OURSYS*	SF0P000			

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SFDD72A	OURSPC*	OURSYS*				
SFDD72B	OURSPC*	OURSYS*				
SFDD72C	OURSPC*	OURSYS*				
SFDD72D	OURSPC*	OURSYS*				
SFDD72E	OURSPC*	OURSYS*				
SFDD72F	OURSPC*	OURSYS*				
SFDD72G	OURSPC*	OURSYS*				
SFDD72H	OURSPC*	OURSYS*				
SFDD72I	OURSPC*	OURSYS*				
SFDDA01	OURSPC*	OURSYS*	SF0P000			
SFDDA02	OURSPC*	OURSYS*	SF0P000			
SFDE220	OURSYS*	OURDMP*	SF0P000	SFDE000	SFDE201	SFDE202
	SFDE203	SFDE204	SFDE205	SFDE206	SFDE207	SFDE208
	SFDE209	SFDE20A	SFDE211	SFDE212	SFDE213	SFDE214
	SFDE215	SFDE216				
SFDE250	OURSYS*	OURDMP*	SF0P000	SFDE000	SFDE231	SFDE232
	SFDE233	SFDE234	SFDE235	SFDE236	SFDE237	SFDE238
	SFDE239	SFDE23A	SFDE241	SFDE242	SFDE243	SFDE244
	SFDE245	SFDE246				
SFDE420	OURSYS*	OURDMP*	SF0P000	SFDE000	SFDE401	SFDE402
	SFDE403	SFDE404	SFDE405	SFDE406	SFDE407	SFDE408
	SFDE409	SFDE40A	SFDE411	SFDE412	SFDE413	SFDE414
	SFDE415	SFDE416				
SFDE450	OURSYS*	OURDMP*	SF0P000	SFDE000	SFDE431	SFDE432
	SFDE433	SFDE434	SFDE435	SFDE436	SFDE437	SFDE438
	SFDE439	SFDE43A	SFDE441	SFDE442	SFDE443	SFDE444
	SFDE445	SFDE446				
SFDF000	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SFDF001	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SFDF002	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SFDF003	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SFDF004	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SFDF005	OURSYS*	OURDMP*	MATHFUN*	SF0P000		
SGD0001	OURSYS*	MATHFUN*	SG0P000	SGD0000		
SGD0003	OURSYS*	SG0P000	SGD0002			
SGD0009	OURSYS*	SG0P000	SGD0004	SGD0005	SGD0006	SGD0007
	SGD0008					
SG0000C	OURSYS*	OURDMP*	SG0P000			
SGD000E	OURSYS*	MATHFUN*	SG0P000	SGD0000	SGD000D	
SGN000K	OURSYS*	SGN000F	SGN000G	SGN000H	SGN000I	SGN000J
SGD0039	OURSPC*	OURSYS*	SG0P000	SGD0004	SGD0005	SGD0006
	SGD0007	SGD0008	SGD0019	SGD0029		
SGD0049	OURSPC*	OURSYS*	SG0P000	SGD0004	SGD0005	SGD0006
	SGD0007	SGD0008	SGD0019	SGD0029		
SGD0059##	OURSPC*	OURSYS*	SG0P000	SGD0004	SGD0005	SGD0006
	SGD0007	SGD0008	SGD0019	SGD0029		
SL09101	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	
SL09111	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	
SL09121	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	
SL09131	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	
SLD9202	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9203	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9204##	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9212	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9213	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9214##	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9222	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9223	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9224##	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9232	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9233	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLD9234##	OURSPC*	OURSYS*	SL09000	SL09100	SL0P000	SLD9200
SLDE2C1	OURSPC*	OURSYS*	SL0P000	SLDE000	SLDE2B1	
SLDE2C2	OURSPC*	OURSYS*	SL0P000	SLDE000	SLDE2B1	

Source file supplied with DEC file type only

Table G-4. Ada Type S Test Programs and Source Code Files (Continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
SLDE2C3##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B1	
SLDE2C4##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B1	
SLDE2C5	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B2	
SLDE2C6	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B2	
SLDE2C7##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B2	
SLDE2C8##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE2B2	
SLDE4C1	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B1	
SLDE4C2	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B1	
SLDE4C3##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B1	
SLDE4C4##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B1	
SLDE4C5	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B2	
SLDE4C6	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B2	
SLDE4C7##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B2	
SLDE4C8##	OURSPC*	OURSYS*	SLOP000	SLDE000	SLDE4B2	
S000104	OURSYS*	OURDMP*	S000002	S000102	S000103	S00P000
S000109	OURSYS*	OURDMP*	S000001	S000107	S000108	S00P000
S000114	OURSYS*	OURDMP*	S000001	S000112	S000113	S00P000
S000204	OURSYS*	OURDMP*	S000202	S000203	S00P000	
S000209	OURSYS*	OURDMP*	S000001	S000207	S000208	S00P000
S000300	OURSYS*	S00P000				
S000305	OURSYS*	S00P000				
S000310	OURSYS*	S00P000				
S000313	OURSYS*	OURDMP*	S000311	S000312	S00P000	
S000316	OURSYS*	OURDMP*	S000314	S000315	S00P000	
S000319	OURSYS*	OURDMP*	S000317	S000318	S00P000	
S000322	OURSYS*	OURDMP*	S000320	S000321	S00P000	
S000325	OURSYS*	OURDMP*	S000323	S000324	S00P000	
S000328	OURSYS*	OURDMP*	S000326	S000327	S00P000	
S000331	OURSYS*	OURDMP*	S000329	S000330	S00P000	
S000334	OURSYS*	OURDMP*	S000332	S000333	S00P000	
S000337	OURSYS*	OURDMP*	S000335	S000336	S00P000	
S000340	OURSYS*	OURDMP*	S000338	S000339	S00P000	
S000343	OURSYS*	OURDMP*	S000341	S000342	S00P000	
S0D0346	OURSYS*	OURDMP*	MATHFUN*	S00P000	S0D0344	S0D0345
S000350	OURSYS*	OURDMP*	S000347	S000348	S000349	S00P000
S000353	OURSYS*	OURDMP*	S000347	S000351	S000352	S00P000
S000356	OURSYS*	OURDMP*	S000354	S000355	S00P000	
S000359	OURSYS*	OURDMP*	S000357	S000358	S00P000	
S000362	OURSYS*	OURDMP*	S000360	S000361	S00P000	
S000365	OURSPC*	OURSYS*	OURDMP*	S000363	S000364	S00P000
S000368	OURSPC*	OURSYS*	OURDMP*	S000366	S000367	S00P000
S00036B	OURSPC*	OURSYS*	OURDMP*	S000369	S00036A	S00P000
S0D036E	OURSPC*	OURSYS*	OURDMP*	S00P000	S0D036C	S0D036D
S00036H	OURSPC*	OURSYS*	OURDMP*	S00036F	S00036G	S00P000
S00036K	OURSPC*	OURSYS*	OURDMP*	S00036I	S00036J	S00P000
S00036N	OURSPC*	OURSYS*	OURDMP*	S00036L	S00036M	S00P000
S0D036Q	OURSPC*	OURSYS*	OURDMP*	S00P000	S0D036O	S0D036P
S0D036T	OURSPC*	OURSYS*	OURDMP*	S00P000	S0D036R	S0D036S
S00036W	OURSPC*	OURSYS*	OURDMP*	S00036U	S00036V	S00P000
S0D036Z	OURSYS*	OURDMP*	OURTYP*	S00P000	S0D036X	S0D036Y
S0D0373	OURSPC*	OURSYS*	OURDMP*	S00P000	S0D0370	S0D0371
	S0D0372					
S000378	OURSYS*	OURDMP*	S000374	S000375	S000376	S000377
	S00P000					
S0D0383	OURSPC*	OURSYS*	OURDMP*	S000374	S00P000	S0D0380
	S0D0381	S0D0382				
S000391	OURSYS*	OURDMP*	S000390	S00P000		
S000404	OURSYS*	OURDMP*	S000402	S000403	S00P000	
S000500	OURSYS*	S00P000				
S000504	OURSYS*	OURDMP*	S000502	S000503	S00P000	
S000509	OURSYS*	OURDMP*	S000507	S000508	S00P000	
S000514	OURSYS*	OURDMP*	S000512	S000513	S00P000	
S000519	OURSYS*	OURDMP*	S000517	S000518	S00P000	
S0D0525	OURSYS*	OURDMP*	S00P000	S0D0523	S0D0524	

Source file supplied with DEC file type only

Table G-4. Ada Type S Test Programs and Source Code Files (Concluded)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
SOD0529	OURSYS*	OURDMP*	SODP000	SOD0526	SOD0527	SOD0528
S000604	OURSYS*	OURDMP*	S000602	S000603	S00P000	
S000605	OURSYS*	OURDMP*	S00P000			
S000606	OURSYS*	OURDMP*	S00P000			
S000607	OURSYS*	OURDMP*	S00P000			
S000704	OURSYS*	OURDMP*	S000702	S000703	S00P000	
S000705	OURSYS*	OURDMP*	S00P000			
S000708	OURSYS*	OURDMP*	S000706	S000707	S00P000	
S000711	OURSYS*	OURDMP*	S000709	S000710	S00P000	
S000713	OURSYS*	OURDMP*	S00P000			
S000714	OURSYS*	OURDMP*	S00P000			
S000715	OURSYS*	OURDMP*	S00P000			
S000716	OURSYS*	OURDMP*	S00P000			
S000717	OURSYS*	OURDMP*	S00P000			
S000718	OURSYS*	OURDMP*	S00P000			
S000721	OURSYS*	OURDMP*	S000719	S000720	S00P000	

Table G-5 Ada Type T test Programs and Source Code Files

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TA00000	OURSYS*	OURDMP*	TA0P000			
TF03519	OURSYS*	OURDMP*	TF03500	TF03501	TF03502	TF03503
	TF03504	TF03509	TF03510	TF03511	TF03512	TF03513
	TF03514	TF03517	TF03518	TF0P000		
TF03550	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD3551	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF03620	OURSYS*	OURDMP*	TF03600	TF03601	TF03602	TF03603
	TF03604	TF03605	TF03606	TF03607	TF03608	TF03609
	TF03610	TF03611	TF03612	TF03613	TF03614	TF03615
	TF03616	TF03617	TF03618	TF03619	TF0P000	
TF03650	OURSYS*	OURDMP*	TF03630	TF03631	TF03632	TF03633
	TF03634	TF03635	TF03636	TF03637	TF03638	TF03639
	TF03641	TF03642	TF03643	TF03644	TF03645	TF03646
	TF03647	TF03648	TF03649	TF0P000		
TF03704	OURSPC*	OURSYS*	OURDMP*	TF03700	TF03701	TF03702
	TF03703	TF0P000				
TF03805	OURSPC*	OURSYS*	OURDMP*	TF03800	TF03801	TF03802
	TF03803	TF03804	TF0P000			
TF04120	OURSYS*	OURDMP*	TF0P000			
TF04121	OURSYS*	OURDMP*	TF0P000			
TF04122	OURSYS*	OURDMP*	TF0P000			
TF04123	OURSYS*	OURDMP*	TF0P000			
TF04124	OURSYS*	OURDMP*	TF0P000			
TF04125	OURSYS*	OURDMP*	TF0P000			
TF04126	OURSYS*	OURDMP*	TF0P000			
TF04127	OURSYS*	OURDMP*	TF0P000			
TFD4128*	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04129	OURSYS*	OURDMP*	TF0P000			
TFD412A	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0412B	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD412C	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD412D	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04130	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04131	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04132	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04133	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4135	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4136	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4137	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4138	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4139	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD413A	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD413B	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD413C	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0413D	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD413E	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD413G	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0413H	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0413I	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0413J	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0413K	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04310	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04311	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04312	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04510	OURSYS*	OURDMP*	TF0P000			
TF04511	OURSYS*	OURDMP*	TF0P000			
TF04512	OURSYS*	OURDMP*	TF0P000			
TF04513	OURSYS*	OURDMP*	TF0P000			
TF04514	OURSYS*	OURDMP*	TF0P000			
TF04515	OURSYS*	OURDMP*	TF0P000			
TF04516	OURSYS*	OURDMP*	TF0P000			
TF04517	OURSYS*	OURDMP*	TF0P000			
TF0451J	OURSYS*	OURDMP*	TF04518	TF04519	TF0451A	TF0451B
	TF0451C	TF0451D	TF0451E	TF0451F	TF0451G	TF0451H
	TF0451I	TF0P000				

* Source file supplied with USE file type only

Table G-5 Ada Type T test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TFD451U†	OURSYS*	OURDMP*	OURTYP*	TF0P000	TFD451K	TFD451L
	TFD451M	TFD451N	TFD451O	TFD451P	TFD451Q	TFD451R
	TFD451S					
TF0451Z	OURSYS*	OURDMP*	TF0451V	TF0451W	TF0451X	TF0451Y
	TF0P000					
TF04520	OURSYS*	OURDMP*	TF0P000			
TF04521	OURSYS*	OURDMP*	TF0P000			
TF04522	OURSYS*	OURDMP*	TF0P000			
TFD4523	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4524	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4525	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4526	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF04527	OURSYS*	OURDMP*	TF0P000			
TFD4528	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04529	OURSYS*	OURDMP*	TF0P000			
TFD452A	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF04530	OURSYS*	OURDMP*	TF0P000			
TF04531	OURSYS*	OURDMP*	TF0P000			
TF04532	OURSYS*	OURDMP*	TF0P000			
TF04533	OURSYS*	OURDMP*	TF0P000			
TF04534	OURSYS*	OURDMP*	TF0P000			
TF04535	OURSYS*	OURDMP*	TF0P000			
TF04536	OURSYS*	OURDMP*	TF0P000			
TF04537	OURSYS*	OURDMP*	TF0P000			
TF04538	OURSYS*	OURDMP*	TF0P000			
TF04539	OURSYS*	OURDMP*	TF0P000			
TF0453A	OURSYS*	OURDMP*	TF0P000			
TF0453B	OURSYS*	OURDMP*	TF0P000			
TF0453C	OURSYS*	OURDMP*	TF0P000			
TF04540	OURSYS*	OURDMP*	TF0P000			
TF04541	OURSYS*	OURDMP*	TF0P000			
TF04550	OURSYS*	OURDMP*	TF0P000			
TF04551	OURSYS*	OURDMP*	TF0P000			
TF04552	OURSYS*	OURDMP*	TF0P000			
TF04553	OURSYS*	OURDMP*	TF0P000			
TF04554	OURSYS*	OURDMP*	TF0P000			
TF04555	OURSYS*	OURDMP*	TF0P000			
TF04556	OURSYS*	OURDMP*	TF0P000			
TF04557	OURSYS*	OURDMP*	TF0P000			
TF04558	OURSYS*	OURDMP*	TF0P000			
TF04559	OURSYS*	OURDMP*	TF0P000			
TFD455D	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455E	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455F	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455G	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455I	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455J	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455K	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD455L	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF0455N	OURSYS*	OURDMP*	TF0P000			
TF0455O	OURSYS*	OURDMP*	TF0P000			
TF04560	OURSYS*	OURDMP*	TF0P000			
TF04562	OURSYS*	OURDMP*	TF0P000			
TFD4566	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4567	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4568	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4569	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF04600	OURSYS*	OURDMP*	TF0P000			
TF04601	OURSYS*	OURDMP*	TF0P000			
TF04602	OURSYS*	OURDMP*	TF0P000			
TFD4603	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4604	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4605	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4606	OURSYS*	OURDMP*	OURTYP*	TF0P000		

† Source file supplied with USE file type only

Table G-5 Ada Type T test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TFD4607	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4608	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD4609	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD460A	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD460B	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF0460E	OURSYS*	OURDMP*	TF0P000			
TF0460F	OURSYS*	OURDMP*	TF0P000			
TF0460G	OURSYS*	OURDMP*	TF0P000			
TF0460H	OURSYS*	OURDMP*	TF0P000			
TF0460I	OURSYS*	OURDMP*	TF0P000			
TF0460J	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD460K	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD460L	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0460M	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD460N	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD460O	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD4804	OURSPC*	OURSYS*	OURDMP*	TF0P000	TFD4800	TFD4801
	TFD4802	TFD4803**				
TF05200	OURSYS*	OURDMP*	TF0P000			
TF05201	OURSYS*	OURDMP*	TF0P000			
TF05202	OURSYS*	OURDMP*	TF0P000			
TF05203	OURSYS*	OURDMP*	TF0P000			
TF05204	OURSYS*	OURDMP*	TF0P000			
TF05205	OURSYS*	OURDMP*	TF0P000			
TF05206	OURSYS*	OURDMP*	TF0P000			
TF05207	OURSYS*	OURDMP*	TF0P000			
TF05208	OURSYS*	OURDMP*	TF0P000			
TF05209	OURSYS*	OURDMP*	TF0P000			
TF0520A	OURSYS*	OURDMP*	TF0P000			
TF0520B	OURSYS*	OURDMP*	TF0P000			
TFD520C	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520D	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520E	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520F	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520G	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520H	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520I	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD520J	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF0520M	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD520N	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0520O	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD520P	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TFD520Q	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0520R	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0520S	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF05210	OURSYS*	OURDMP*	TF0P000			
TF05211	OURSYS*	OURDMP*	TF0P000			
TF05212	OURSYS*	OURDMP*	TF0P000			
TF05213	OURSYS*	OURDMP*	TF0P000			
TF05214	OURSYS*	OURDMP*	TF0P000			
TF05215	OURSYS*	OURDMP*	TF0P000			
TFD5218	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD5219	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521A	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521B	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521C	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521D	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521E	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TFD521F	OURSYS*	OURDMP*	OURTYP*	TF0P000		
TF0521H	OURSYS*	OURDMP*	TF0P000			
TF0521I	OURSYS*	OURDMP*	TF0P000			
TF0521J	OURSYS*	OURDMP*	TF0P000			
TF0521K	OURSYS*	OURDMP*	TF0P000			
TFD521K	OURSPC*	OURSYS*	OURDMP*	TF0P000		

** Multiple versions of source file supplied(USE and ADA file types)

Table G-5 Ada Type T test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
TF0521L	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0521M	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0521N	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0521O	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF0521P	OURSPC*	OURSYS*	OURDMP*	TF0P000		
TF05304	OURSYS*	OURDMP*	TF05300	TF05301	TF05302	TF05303
TF05308	TF0P000					
TF05308	OURSYS*	OURDMP*	TF05305	TF05306	TF05307	TF0P000
TF05408	OURSYS*	OURDMP*	TF05400	TF05401	TF05402	TF05403
	TF05404	TF05405	TF05406	TF05407	TF0P000	
TF05505	OURSYS*	OURDMP*	TF05501	TF05502	TF05503	TF05504
	TF0P000					
TF0550C	OURSYS*	OURDMP*	TF05506	TF05507	TF05508	TF05509
	TF0550A	TF0550B	TF0P000			
TF0550D	OURSYS*	OURDMP*	TF0P000			
TF0550E	OURSYS*	OURDMP*	TF0P000			
TF0550F	OURSYS*	OURDMP*	TF0P000			
TF06001	OURSYS*	OURDMP*	TF0P000			
TF06009	OURSYS*	OURDMP*	TF0P000			
TF06010	OURSYS*	OURDMP*	TF0P000			
TF06011	OURSYS*	OURDMP*	TF0P000			
TF06022	OURSYS*	OURDMP*	TF06013	TF06014	TF06015	TF06016
	TF06017	TF06018	TF06019	TF0P000		
TF06033	OURSYS*	OURDMP*	TF06023	TF06024	TF06025	TF06026
	TF06027	TF06028	TF06029	TF06030	TF06031	TF06032
	TF0P000					
TF06053	OURSYS*	OURDMP*	TF06043	TF06044	TF06045	TF06046
	TF06047	TF06048	TF06049	TF06050	TF06051	TF06052
	TF0P000					
TF06069	OURSYS*	OURDMP*	TF06060	TF06061	TF06062	TF06063
	TF06064	TF06065	TF06066	TF06067	TF06068	TF0P000
TF06079	OURSYS*	OURDMP*	TF06070	TF06071	TF06072	TF06073
	TF06074	TF06075	TF06076	TF06077	TF06078	TF0P000
TF06101	OURSYS*	OURDMP*	TF06100	TF0P000		
TF06109	OURSYS*	OURDMP*	TF06108	TF0P000		
TF06110	OURSYS*	OURDMP*	TF06108	TF0P000		
TF06111	OURSYS*	OURDMP*	TF06108	TF0P000		
TF06122	OURSYS*	OURDMP*	TF06112	TF06113	TF06114	TF06115
	TF06116	TF06117	TF06118	TF06119	TF0P000	
TF06140	OURSYS*	OURDMP*	TF06132	TF06133	TF06134	TF06135
	TF06136	TF06137	TF06138	TF06139	TF0P000	
TF06150	OURSYS*	OURDMP*	TF06142	TF06143	TF06144	TF06145
	TF06146	TF06147	TF06148	TF06149	TF0P000	
TF06160	OURSYS*	OURDMP*	TF06152	TF06153	TF06154	TF06155
	TF06156	TF06157	TF06158	TF06159	TF0P000	
TF06170	OURSYS*	OURDMP*	TF06162	TF06163	TF06164	TF06165
	TF06166	TF06167	TF06168	TF06169	TF0P000	
TF06180	OURSYS*	OURDMP*	TF06172	TF06173	TF06174	TF06175
	TF06176	TF06177	TF06178	TF06179	TF0P000	
TF06190	OURSYS*	OURDMP*	TF06182	TF06183	TF06184	TF06185
	TF06186	TF06187	TF06188	TF06189	TF0P000	
TF06199	OURSPC*	OURSYS*	OURDMP*	TF06191	TF06192	TF06193
	TF06194	TF06195	TF06196	TF06197	TF06198	TF0P000
TF0619I	OURSPC*	OURSYS*	OURDMP*	TF0619A	TF0619B	TF0619C
	TF0619D	TF0619E	TF0619F	TF0619G	TF0619H	TF0P000
TF0619R	OURSPC*	OURSYS*	OURDMP*	TF0619J	TF0619K	TF0619L
	TF0619M	TF0619N	TF0619O	TF0619P	TF0619Q	TF0P000
TFD6201	OURSYS*	OURDMP*	TF0P000			
TFD6209	OURSYS*	OURDMP*	TF0P000			
TFD6210	OURSYS*	OURDMP*	TF0P000			
TFD6211	OURSYS*	OURDMP*	TF0P000			
TFD6222	OURSYS*	OURDMP*	TF0P000			
	TFD6216	TFD6217	TFD6218	TFD6213	TFD6214	TFD6215
TFD6233	OURSYS*	OURDMP*	TF0P000	TFD6223	TFD6224	TFD6225
	TFD6226	TFD6227	TFD6228	TFD6229	TFD6230	TFD6231

Table G-5 Ada Type T test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
TFD6253	TFD6232 OURSYS*	OURDMP*	TFOP000	TFD6243	TFD6244	TFD6245
	TFD6246	TFD6247	TFD6248	TFD6249	TFD6250	TFD6251
	TFD6252					
TFD6269	OURSYS*	OURDMP*	TFOP000	TFD6260	TFD6261	TFD6262
	TFD6263	TFD6264	TFD6265	TFD6266	TFD6267	TFD6268
TFD6279	OURSYS*	OURDMP*	TFOP000	TFD6270	TFD6271	TFD6272
	TFD6273	TFD6274	TFD6275	TFD6276	TFD6277	TFD6278
TFD6301	OURSYS*	OURDMP*	TFOP000	TFD6300		
TFD6309	OURSYS*	OURDMP*	TFOP000	TFD6308		
TFD6310	OURSYS*	OURDMP*	TFOP000	TFD6308		
TFD6311	OURSYS*	OURDMP*	TFOP000	TFD6308		
TFD6322	OURSYS*	OURDMP*	TFOP000	TFD6312	TFD6313	TFD6314
	TFD6315	TFD6316	TFD6317	TFD6318	TFD6319	
TFD6340	OURSYS*	OURDMP*	TFOP000	TFD6332	TFD6333	TFD6334
	TFD6335	TFD6336	TFD6337	TFD6338	TFD6339	
TFD6350	OURSYS*	OURDMP*	TFOP000	TFD6342	TFD6343	TFD6344
	TFD6345	TFD6346	TFD6347	TFD6348	TFD6349	
TFD6360	OURSYS*	OURDMP*	TFOP000	TFD6352	TFD6353	TFD6354
	TFD6355	TFD6356	TFD6357	TFD6358	TFD6359	
TFD6370	OURSYS*	OURDMP*	TFOP000	TFD6362	TFD6363	TFD6364
	TFD6365	TFD6366	TFD6367	TFD6368	TFD6369	
TFD6380	OURSYS*	OURDMP*	TFOP000	TFD6372	TFD6373	TFD6374
	TFD6375	TFD6376	TFD6377	TFD6378	TFD6379	
TFD6390	OURSYS*	OURDMP*	TFOP000	TFD6382	TFD6383	TFD6384
	TFD6385	TFD6386	TFD6387	TFD6388	TFD6389	
TF0642B	OURSPC*	OURSYS*	OURDMP*	TF06422	TF06423	TF06424
	TF06425	TF06426	TF06427	TF06428	TF06429	TF0642A
	TFOP000					
TF06803	OURSYS*	OURDMP*	TF06802	TFOP000		
TF06807	OURSYS*	OURDMP*	TF06806	TFOP000		
TF06809	OURSPC*	OURSYS*	OURDMP*	TF06808	TFOP000	
TF0680B	OURSPC*	OURSYS*	OURDMP*	TF0680A	TFOP000	
TF0680D	OURSPC*	OURSYS*	OURDMP*	TF0680C	TFOP000	
TF0680F	OURSPC*	OURSYS*	OURDMP*	TF0680E	TFOP000	
TFD680H	OURSPC*	OURSYS*	OURDMP*	TFOP000	TFD680G	
TFD680J	OURSPC*	OURSYS*	OURDMP*	TFOP000	TFD680I	
TF06811	OURSYS*	OURDMP*	TF06810	TFOP000		
TF06815	OURSYS*	OURDMP*	TF06814	TFOP000		
TF06817	OURSYS*	OURDMP*	TF06816	TFOP000		
TF06819	OURSYS*	OURDMP*	TF06818	TFOP000		
TF06821	OURSYS*	OURDMP*	TF06820	TFOP000		
TF06823	OURSPC*	OURSYS*	OURDMP*	TF06822	TFOP000	
TF06825	OURSPC*	OURSYS*	OURDMP*	TF06824	TFOP000	
TFN9301	OURSPC*	OURSYS*	TFN9300			
TFN9302	OURSPC*	OURSYS*	TFN9300			
TFN9303	OURSPC*	OURSYS*	TFN9300			
TF09501	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09502	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09503	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09504	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09505	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09506	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09507	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09508	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TF09509	OURSPC*	OURSYS*	OURDMP*	TF09500	TFOP000	
TFN9511	OURSPC*	OURSYS*	TFN9510			
TF09600	OURSPC*	OURSYS*	TFOP000			
TF09601	OURSPC*	OURSYS*	TFOP000			
TF09602	OURSPC*	OURSYS*	TFOP000			
TF09603	OURSPC*	OURSYS*	TFOP000			
TF09604	OURSPC*	OURSYS*	TFOP000			
TF09605	OURSPC*	OURSYS*	TFOP000			
TF09606	OURSPC*	OURSYS*	TFOP000			
TF09607	OURSPC*	OURSYS*	TFOP000			

Table G-5 Ada Type T test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TFN9611	OURSPC*	OURSYS*	TFN9610			
TFN9612	OURSPC*	OURSYS*	TFN9610			
TFN9613	OURSPC*	OURSYS*	TFN9610			
TF09711	OURSYS*	TF09710	TF0P000			
TF09713	OURSYS*	TF09712	TF0P000			
TF09720	OURSPC*	OURSYS*	OURDMP*	TF09500	TF0P000	
TF09721	OURSPC*	OURSYS*	OURDMP*	TF09500	TF0P000	
TF09731	OURSPC*	OURSYS*	OURDMP*	TF09500	TF0P000	
TF09901	OURSYS*	TF09900	TF0P000			
TF09902	OURSYS*	TF09900	TF0P000			
TF09903	OURSYS*	TF09900	TF0P000			
TFM9A01	OURSPC*	OURSYS*	TFM9A00			
TFM9A02	OURSPC*	OURSYS*	TFM9A00			
TFM9A03	OURSPC*	OURSYS*	TFM9A00			
TF09B01	OURSPC*	OURSYS*	OURDMP*	TF09500	TF0P000	
TFD9C00	OURSPC*	OURSYS*	TF0P000	TFD9000		
TFD9C01	OURSPC*	OURSYS*	TF0P000	TFD9000	TFD9200	
TFD9C02	OURSPC*	OURSYS*	TF0P000	TFD9000	TFD9200	
TFD9C03**	OURSPC*	OURSYS*	TF0P000	TFD9000	TFD9200	
TFD9C04**	OURSPC*	OURSYS*	TF0P000	TFD9000	TFD9200	
TFMB001	OURSYS*	TFMB000				
TFMB003	OURSYS*	TFMB002				
TFMB005	OURSYS*	TFMB004				
TFMB007	OURSYS*	TFMB006				
TFMB009	OURSYS*	TFMB008				
TFMB012	OURSYS*	TFMB010	TFMB011			
TFMB016	OURSYS*	TFMB013	TFMB014	TFMB015		
TFMB020	OURSYS*	TFMB017	TFMB018	TFMB019		
TFMB024	OURSYS*	TFMB021	TFMB022	TFMB023		
TFMB028	OURSYS*	TFMB025	TFMB026	TFMB027		
TFMB032	OURSYS*	TFMB013	TFMB014	TFMB015		
TFMB036	OURSYS*	TFMB033	TFMB034	TFMB035		
TFMB040	OURSYS*	TFMB037	TFMB038	TFMB039		
TFMB044	OURSYS*	TFMB041	TFMB042	TFMB043		
TF0C301	OURSYS*	OURDMP*	TF0C100	TF0C101	TF0C300	TF0P000
TF0C302	OURSYS*	OURDMP*	TF0C100	TF0C101	TF0C300	TF0P000
TF0C304	OURSYS*	OURDMP*	TF0C102	TF0C103	TF0C303	TF0P000
TF0C305	OURSYS*	OURDMP*	TF0C102	TF0P000		
TF0C306	OURSYS*	OURDMP*	TF0C102	TF0P000		
TF0C307	OURSYS*	OURDMP*	TF0C102	TF0P000		
TFDC311	OURSYS*	OURDMP*	TF0P000	TFDC104	TFDC105	TFDC310
TFDC314	OURSYS*	OURDMP*	TF0P000	TFDC106	TFDC107**	TFDC313
TFDD600	OURSPC*	OURSYS*	TF0P000			
TFDD601	OURSPC*	OURSYS*	TF0P000			
TFDD602	OURSPC*	OURSYS*	TF0P000			
TFDD603	OURSPC*	OURSYS*	TF0P000			
TFDD604	OURSPC*	OURSYS*	TF0P000			
TFDD605	OURSPC*	OURSYS*	TF0P000			
TFDD606	OURSPC*	OURSYS*	TF0P000			
TFDD607	OURSPC*	OURSYS*	TF0P000			
TFDD608*	OURSPC*	OURSYS*	TF0P000			
TFDD609	OURSPC*	OURSYS*	TF0P000			
TFDD610	OURSPC*	OURSYS*	TF0P000			
TFDD611	OURSPC*	OURSYS*	TF0P000			
TF0D720	OURSYS*					
TF0D721	OURSYS*					
TFDD722	OURSYS*	OURTYP*				
TFDD723	OURSYS*	OURTYP*				
TFDD724	OURSYS*	OURTYP*				
TFDD725	OURSYS*	OURTYP*				
TF0D727	OURSYS*					
TF0D728	OURSPC*	OURSYS*				
TF0D729	OURSPC*	OURSYS*				

** Multiple versions of source file supplied(USE and ADA file types)

* Source file supplied with DEC file type only

* Source file supplied with USE file type only

Table G-5 Ada Type T test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TFDD72A	OURSPC*	OURSYS*				
TFDD72B	OURSPC*	OURSYS*				
TFDD72C	OURSPC*	OURSYS*				
TFDD72D	OURSPC*	OURSYS*				
TFDD72E	OURSPC*	OURSYS*				
TFDD72F	OURSPC*	OURSYS*				
TFDD72G	OURSPC*	OURSYS*				
TFDD72H	OURSPC*	OURSYS*				
TFDD72I	OURSPC*	OURSYS*				
TFDDA01	OURSPC*	OURSYS*	TFOP000			
TFDDA02	OURSPC*	OURSYS*	TFOP000			
TFDE220	OURSYS*	OURDMP*	TFOP000	TFDE000	TFDE201	TFDE202
	TFDE203	TFDE204	TFDE205	TFDE206	TFDE207	TFDE208
	TFDE209	TFDE20A	TFDE211	TFDE212	TFDE213	TFDE214
	TFDE215	TFDE216				
TFDE250	OURSYS*	OURDMP*	TFOP000	TFDE000	TFDE231	TFDE232
	TFDE233	TFDE234	TFDE235	TFDE236	TFDE237	TFDE238
	TFDE239	TFDE23A	TFDE241	TFDE242	TFDE243	TFDE244
	TFDE245	TFDE246				
TFDE420	OURSYS*	OURDMP*	TFOP000	TFDE000	TFDE401	TFDE402
	TFDE403	TFDE404	TFDE405	TFDE406	TFDE407	TFDE408
	TFDE409	TFDE40A	TFDE411	TFDE412	TFDE413	TFDE414
	TFDE415	TFDE416				
TFDE450	OURSYS*	OURDMP*	TFOP000	TFDE000	TFDE431	TFDE432
	TFDE433	TFDE434	TFDE435	TFDE436	TFDE437	TFDE438
	TFDE439	TFDE43A	TFDE441	TFDE442	TFDE443	TFDE444
	TFDE445	TFDE446				
TFDF000	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TFDF001	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TFDF002	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TFDF003	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TFDF004	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TFDF005	OURSYS*	OURDMP*	MATHFUN*	TFOP000		
TGD0001	OURSYS*	MATHFUN*	TG0P000	TGD0000		
TGD0003	OURSYS*	TG0P000	TGD0002			
TGD0009	OURSYS*	TG0P000	TGD0004	TGD0005	TGD0006	TGD0007
	TGD0008					
TGD0000C	OURSYS*	OURDMP*	TG0P000			
TGD0000E	OURSYS*	MATHFUN*	TG0P000	TGD0000	TGD0000	
TGN000K	OURSYS*	TGN000F	TGN000G	TGN000H	TGN000I	TGN000J
TGD0039	OURSPC*	OURSYS*	TG0P000	TGD0004	TGD0005	TGD0006
	TGD0007	TGD0008	TGD0019	TGD0029		
TGD0049	OURSPC*	OURSYS*	TG0P000	TGD0004	TGD0005	TGD0006
	TGD0007	TGD0008	TGD0019	TGD0029		
TGD0059**	OURSPC*	OURSYS*	TG0P000	TGD0004	TGD0005	TGD0006
	TGD0007	TGD0008	TGD0019	TGD0029		
TL09101	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	
TL09111	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	
TL09121	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	
TL09131	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	
TL09202	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09203	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09204**	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09212	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09213	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09214**	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09222	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09223	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09224**	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09232	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09233	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TL09234**	OURSPC*	OURSYS*	TL09000	TL09100	TL0P000	TL09200
TLDE2C1	OURSPC*	OURSYS*	TL0P000	TLDE000	TLDE201	
TLDE2C2	OURSPC*	OURSYS*	TL0P000	TLDE000	TLDE201	

** Source file supplied with DEC file type only

Table G-5 Ada Type T test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)					
TLDE2C3**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B1	
TLDE2C4**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B1	
TLDE2C5	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B2	
TLDE2C6	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B2	
TLDE2C7**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B2	
TLDE2C8**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE2B2	
TLDE4C1	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B1	
TLDE4C2	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B1	
TLDE4C3**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B1	
TLDE4C4**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B1	
TLDE4C5	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B2	
TLDE4C6	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B2	
TLDE4C7**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B2	
TLDE4C8**	OURSPC*	OURSYS*	TLOP000	TLDE000	TLDE4B2	
T000104	OURSYS*	OURDMP*	T000002	T000102	T000103	T00P000
T000109	OURSYS*	OURDMP*	T000001	T000107	T000108	T00P000
T000114	OURSYS*	OURDMP*	T000001	T000112	T000113	T00P000
T000204	OURSYS*	OURDMP*	T000202	T000203	T00P000	
T000209	OURSYS*	OURDMP*	T000001	T000207	T000208	T00P000
T000300	OURSYS*	T00P000				
T000305	OURSYS*	T00P000				
T000310	OURSYS*	T00P000				
T000313	OURSYS*	OURDMP*	T000311	T000312	T00P000	
T000316	OURSYS*	OURDMP*	T000314	T000315	T00P000	
T000319	OURSYS*	OURDMP*	T000317	T000318	T00P000	
T000322	OURSYS*	OURDMP*	T000320	T000321	T00P000	
T000325	OURSYS*	OURDMP*	T000323	T000324	T00P000	
T000328	OURSYS*	OURDMP*	T000326	T000327	T00P000	
T000331	OURSYS*	OURDMP*	T000329	T000330	T00P000	
T000334	OURSYS*	OURDMP*	T000332	T000333	T00P000	
T000337	OURSYS*	OURDMP*	T000335	T000336	T00P000	
T000340	OURSYS*	OURDMP*	T000338	T000339	T00P000	
T000343	OURSYS*	OURDMP*	T000341	T000342	T00P000	
T000346	OURSYS*	OURDMP*	MATHFUN*	T00P000	T000344	T000345
T000350	OURSYS*	OURDMP*	T000347	T000348	T000349	T00P000
T000353	OURSYS*	OURDMP*	T000347	T000351	T000352	T00P000
T000356	OURSYS*	OURDMP*	T000354	T000355	T00P000	
T000359	OURSYS*	OURDMP*	T000357	T000358	T00P000	
T000362	OURSYS*	OURDMP*	T000360	T000361	T00P000	
T000365	OURSPC*	OURSYS*	OURDMP*	T000363	T000364	T00P000
T000368	OURSPC*	OURSYS*	OURDMP*	T000366	T000367	T00P000
T00036B	OURSPC*	OURSYS*	OURDMP*	T000369	T00036A	T00P000
T00036E	OURSPC*	OURSYS*	OURDMP*	T00P000	T00036C	T00036D
T00036H	OURSPC*	OURSYS*	OURDMP*	T00036F	T00036G	T00P000
T00036K	OURSPC*	OURSYS*	OURDMP*	T00036I	T00036J	T00P000
T00036N	OURSPC*	OURSYS*	OURDMP*	T00036L	T00036M	T00P000
T00036Q	OURSPC*	OURSYS*	OURDMP*	T00P000	T00036O	T00036P
T00036T	OURSPC*	OURSYS*	OURDMP*	T00P000	T00036R	T00036S
T00036W	OURSPC*	OURSYS*	OURDMP*	T00036U	T00036V	T00P000
T00036Z	OURSYS*	OURDMP*	OURTYP*	T00P000	T00036X	T00036Y
T000373	OURSPC*	OURSYS*	OURDMP*	T00P000	T000370	T000371
	T000372					
T000378	OURSYS*	OURDMP*	T000374	T000375	T000376	T000377
	T00P000					
T000383	OURSPC*	OURSYS*	OURDMP*	T000374	T00P000	T000380
	T000381	T000382				
T000391	OURSYS*	OURDMP*	T000390	T00P000		
T000404	OURSYS*	OURDMP*	T000402	T000403	T00P000	
T000500	OURSYS*	T00P000				
T000504	OURSYS*	OURDMP*	T000502	T000503	T00P000	
T000509	OURSYS*	OURDMP*	T000507	T000508	T00P000	
T000514	OURSYS*	OURDMP*	T000512	T000513	T00P000	
T000519	OURSYS*	OURDMP*	T000517	T000518	T00P000	
T000525	OURSYS*	OURDMP*	T00P000	T000523	T000524	

** Source file supplied with DEC file type only

Table G-5 Ada Type T test Programs and Source Code Files (concluded)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - TEST SUPPORT SOFTWARE PACKAGES USED)</u>					
TOD0529	OURSYS*	OURDMP*	T00P000	TOD0526	TOD0527	TOD0528
T000604	OURSYS*	OURDMP*	T000602	T000603	T00P000	
T000605	OURSYS*	OURDMP*	T00P000			
T000606	OURSYS*	OURDMP*	T00P000			
T000607	OURSYS*	OURDMP*	T00P000			
T000704	OURSYS*	OURDMP*	T000702	T000703	T00P000	
T000705	OURSYS*	OURDMP*	T00P000			
T000708	OURSYS*	OURDMP*	T000706	T000707	T00P000	
T000711	OURSYS*	OURDMP*	T000709	T000710	T00P000	
T000713	OURSYS*	OURDMP*	T00P000			
T000714	OURSYS*	OURDMP*	T00P000			
T000715	OURSYS*	OURDMP*	T00P000			
T000716	OURSYS*	OURDMP*	T00P000			
T000717	OURSYS*	OURDMP*	T00P000			
T000718	OURSYS*	OURDMP*	T00P000			
T000721	OURSYS*	OURDMP*	T000719	T000720	T00P000	

Table G-6 JOVIAL Test Programs and Source Code Files

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
JA00000	OURJOV*	JOVDMP*	JA0P000*			
JF03519	OURJOV*	JOVDMP*	JF03500	JF03501	JF03502	JF03503
	JF03504	JF03513	JF03514	JF0P000*	JFJ3500*	JFJ3501*
	JFJ3502*	JFJ3503*	JFJ3504*	JFJ3513*	JFJ3514*	
JF03550	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD3551	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF03620	OURJOV*	JOVDMP*	JF03600	JF03601	JF03602	JF03603
	JF03604	JF03605	JF03606	JF03607	JF03608	JF03609
	JF03610	JF0P000*	JFJ3600*	JFJ3601*	JFJ3602*	JFJ3603*
	JFJ3604*	JFJ3605*	JFJ3606*	JFJ3607*	JFJ3608*	JFJ3609*
	JFJ3610*					
JF03704	OURJOV*	JOVDMP*	JOVSPC*	JF03700	JF03701**	JF03702
	JF03703	JF0P000*	JFJ3700*	JFJ3701*	JFJ3702*	JFJ3703*
JF03805	OURJOV*	JOVDMP*	JOVSPC*	JF03800	JF03801	JF03802
	JF03803	JF03804	JF0P000*	JFJ3800*	JFJ3801*	JFJ3802*
	JFJ3803*	JFJ3804*				
JF04120	OURJOV*	JOVDMP*	JF0P000*			
JF04121	OURJOV*	JOVDMP*	JF0P000*			
JF04122	OURJOV*	JOVDMP*	JF0P000*			
JF04123	OURJOV*	JOVDMP*	JF0P000*			
JF04124	OURJOV*	JOVDMP*	JF0P000*			
JF04125	OURJOV*	JOVDMP*	JF0P000*			
JF04126	OURJOV*	JOVDMP*	JF0P000*			
JF04127	OURJOV*	JOVDMP*	JF0P000*			
JFD4128	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04129	OURJOV*	JOVDMP*	JF0P000*			
JFD412A	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0412B	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD412C	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD412D	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04130	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04131	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04132	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04133	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD4135	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD4136	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD4137	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD4138	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD4139	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD413A	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD413B	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD413C	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0413D	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD413E	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JFD413G	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0413H	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0413I	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0413J	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF0413K	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04310	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04311**	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04312	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04510	OURJOV*	JOVDMP*	JF0P000*			
JF04511	OURJOV*	JOVDMP*	JF0P000*			
JF04512	OURJOV*	JOVDMP*	JF0P000*			
JF04513	OURJOV*	JOVDMP*	JF0P000*			
JF04514	OURJOV*	JOVDMP*	JF0P000*			
JF04515	OURJOV*	JOVDMP*	JF0P000*			
JF04516	OURJOV*	JOVDMP*	JF0P000*			
JF04517	OURJOV*	JOVDMP*	JF0P000*			
JF0451J**	OURJOV*	JOVDMP*	JF04518	JF04519	JF0451A	JF0451B
	JF0451C	JF0451D	JF0451E	JF0451F	JF0451G	JF0451H*

* Source file supplied with USE file type only

** Multiple versions of source file supplied(USE and JOV/CPL file types)

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
	JF0451I*	JF0P000*	JFJ4518*	JFJ4519*	JFJ451A*	JFJ451B*
	JFJ451C*	JFJ451D*	JFJ451E*	JFJ451F*	JFJ451G*	JFJ451H*
	JFJ451I*					
JFD451U*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*	JFD451K	JFD451L
	JFD451M	JFD451N*	JFD451O*	JFD451P	JFD451Q	JFD451R*
	JFD451S*	JFJ451K*	JFJ451L*	JFJ451M*	JFJ451N*	JFJ451O*
	JFJ451P*	JFJ451Q*	JFJ451R*	JFJ451S*	JFJ451T*	
JF0451Z	OURJOV*	JOVDMP*	JF0451V	JF0451W	JF0451X	JF0451Y
	JF0P000*	JFJ451V*	JFJ451W*	JFJ451X*	JFJ451Y*	
JF04520	OURJOV*	JOVDMP*	JF0P000*			
JF04521	OURJOV*	JOVDMP*	JF0P000*			
JF04522	OURJOV*	JOVDMP*	JF0P000*			
JFD4523	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4524	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4525*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4526*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JF04527	OURJOV*	JOVDMP*	JF0P000*			
JFD4528*	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04529	OURJOV*	JOVDMP*	JF0P000*			
JFD452A	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*		
JF04530	OURJOV*	JOVDMP*	JF0P000*			
JF04531	OURJOV*	JOVDMP*	JF0P000*			
JF04532	OURJOV*	JOVDMP*	JF0P000*			
JF04533	OURJOV*	JOVDMP*	JF0P000*			
JF04534	OURJOV*	JOVDMP*	JF0P000*			
JF04535	OURJOV*	JOVDMP*	JF0P000*			
JF04536	OURJOV*	JOVDMP*	JF0P000*			
JF04537	OURJOV*	JOVDMP*	JF0P000*			
JF04538	OURJOV*	JOVDMP*	JF0P000*			
JF04539	OURJOV*	JOVDMP*	JF0P000*			
JF0453A	OURJOV*	JOVDMP*	JF0P000*			
JF0453B	OURJOV*	JOVDMP*	JF0P000*			
JF0453C	OURJOV*	JOVDMP*	JF0P000*			
JF04540	OURJOV*	JOVDMP*	JF0P000*			
JF04541	OURJOV*	JOVDMP*	JF0P000*			
JF04550	OURJOV*	JOVDMP*	JF0P000*			
JF04551	OURJOV*	JOVDMP*	JF0P000*			
JF04552	OURJOV*	JOVDMP*	JF0P000*			
JF04553	OURJOV*	JOVDMP*	JF0P000*			
JF04554	OURJOV*	JOVDMP*	JF0P000*			
JF04555	OURJOV*	JOVDMP*	JF0P000*			
JF04556	OURJOV*	JOVDMP*	JF0P000*			
JF04557	OURJOV*	JOVDMP*	JF0P000*			
JF04558	OURJOV*	JOVDMP*	JF0P000*			
JF04559	OURJOV*	JOVDMP*	JF0P000*			
JFD455D	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455E	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455F*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455G*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455I	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455J	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455K*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD455L*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JF0455N	OURJOV*	JOVDMP*	JF0P000*			
JF0455O	OURJOV*	JOVDMP*	JF0P000*			
JF04560	OURJOV*	JOVDMP*	JF0P000*			
JF04562	OURJOV*	JOVDMP*	JF0P000*			
JF04563	OURJOV*	JOVDMP*	JF0P000*			
JFD4566	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4567	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4568*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JFD4569*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*		
JF04600	OURJOV*	JOVDMP*	JF0P000*			

* Source file supplied with USE file type only

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)</u>			
JF04601	OURJOV*	JOVDMP*	JF0P000*	
JF04602	OURJOV*	JOVDMP*	JF0P000*	
JFD4603	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4604	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4605	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4606	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4607	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4608*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD4609*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD460A*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD460B*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD460C*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD460D*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JF0460E	OURJOV*	JOVDMP*	JF0P000*	
JF0460F	OURJOV*	JOVDMP*	JF0P000*	
JF0460G	OURJOV*	JOVDMP*	JF0P000*	
JF0460H	OURJOV*	JOVDMP*	JF0P000*	
JF0460I	OURJOV*	JOVDMP*	JF0P000*	
JF0460J	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD460K	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD460L	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF0460M	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD460N	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD460O	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF05200	OURJOV*	JOVDMP*	JF0P000*	
JF05201	OURJOV*	JOVDMP*	JF0P000*	
JF05202	OURJOV*	JOVDMP*	JF0P000*	
JF05203	OURJOV*	JOVDMP*	JF0P000*	
JF05204	OURJOV*	JOVDMP*	JF0P000*	
JF05205	OURJOV*	JOVDMP*	JF0P000*	
JF05206	OURJOV*	JOVDMP*	JF0P000*	
JF05207	OURJOV*	JOVDMP*	JF0P000*	
JF05208	OURJOV*	JOVDMP*	JF0P000*	
JF05209	OURJOV*	JOVDMP*	JF0P000*	
JF0520A	OURJOV*	JOVDMP*	JF0P000*	
JF0520B	OURJOV*	JOVDMP*	JF0P000*	
JFD520C	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520D	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520E	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520F	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520G*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520H*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520I*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD520J*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JF0520M	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD520N	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF0520O	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD520P	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JFD520Q	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF0520R	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF0520S	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*
JF05210	OURJOV*	JOVDMP*	JF0P000*	
JF05211	OURJOV*	JOVDMP*	JF0P000*	
JF05212	OURJOV*	JOVDMP*	JF0P000*	
JF05213	OURJOV*	JOVDMP*	JF0P000*	
JF05214	OURJOV*	JOVDMP*	JF0P000*	
JF05215	OURJOV*	JOVDMP*	JF0P000*	
JFD5218	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD5219	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD521A	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD521B	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD521C*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*
JFD521D*	OURJOV*	JOVDMP*	JOVTYP*	JF0P000*

* Source file supplied with USE file type only

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
JFD521E*	OURJOV*	JOVDMP*	JOVTYP*	JFOP000*		
JFD521F*	OURJOV*	JOVDMP*	JOVTYP*	JFOP000*		
JF0521H	OURJOV*	JOVDMP*	JFOP000*			
JF0521I	OURJOV*	JOVDMP*	JFOP000*			
JF0521J	OURJOV*	JOVDMP*	JFOP000*			
JF0521K	OURJOV*	JOVDMP*	JFOP000*			
JFD521K	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JF0521L	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JFD521M	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JF0521N	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JFD521O	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JFD521P	OURJOV*	JOVDMP*	JOVSPC*	JFOP000*		
JF05304	OURJOV*	JOVDMP*	JF05300	JF05301	JF05302	JF05303
	JFOP000*	JFJ5300*	JFJ5301*	JFJ5302*	JFJ5303*	
JF05308	OURJOV*	JOVDMP*	JF05305	JF05306	JF05307	JFOP000*
	JFJ5305*	JFJ5306*	JFJ5307*			
JF05408	OURJOV*	JOVDMP*	JF05400	JF05401	JF05402	JF05403
	JF05404	JF05405	JF05406	JF05407	JFOP000*	JFJ5400*
	JFJ5401*	JFJ5402*	JFJ5403*	JFJ5404*	JFJ5405*	JFJ5406*
	JFJ5407*					
JF05505	OURJOV*	JOVDMP*	JF05501	JF05502	JF05503	JF05504
	JFOP000*	JFJ5501*	JFJ5502*	JFJ5503*	JFJ5504*	
JF0550C	OURJOV*	JOVDMP*	JF05506	JF05507	JF05508	JF05509
	JF0550A	JF0550B	JFOP000*	JFJ5506*	JFJ5507*	JFJ5508*
	JFJ5509*	JFJ550A*	JFJ550B*			
JF0550D	OURJOV*	JOVDMP*	JFOP000*			
JF0550E	OURJOV*	JOVDMP*	JFOP000*			
JF0550F	OURJOV*	JOVDMP*	JFOP000*			
JF06001	OURJOV*	JOVDMP*	JFOP000*			
JF06009	OURJOV*	JOVDMP*	JFOP000*			
JF06010	OURJOV*	JOVDMP*	JFOP000*			
JF06011	OURJOV*	JOVDMP*	JFOP000*			
JF06022	OURJOV*	JOVDMP*	JF06013	JF06014	JF06015	JF06016
	JF06017	JF06018	JF06019	JFOP000*	JFJ6013*	JFJ6014*
	JFJ6015*	JFJ6016*	JFJ6017*	JFJ6018*	JFJ6019*	
JF06033	OURJOV*	JOVDMP*	JF06023	JF06024	JF06025	JF06026
	JF06027	JF06028	JF06029	JF06030	JF06031	JF06032
	JFOP000*	JFJ6023*	JFJ6024*	JFJ6025*	JFJ6026*	JFJ6027*
	JFJ6028*	JFJ6029*	JFJ6030*	JFJ6031*	JFJ6032*	
JF06053	OURJOV*	JOVDMP*	JF06043	JF06044	JF06045	JF06046
	JF06047	JF06048	JF06049	JF06050	JF06051	JF06052
	JFOP000*	JFJ6043*	JFJ6044*	JFJ6045*	JFJ6046*	JFJ6047*
	JFJ6048*	JFJ6049*	JFJ6050*	JFJ6051*	JFJ6052*	
JF06069	OURJOV*	JOVDMP*	JF06060	JF06061	JF06062	JF06063
	JF06064	JF06065	JF06066	JF06067	JF06068	JFOP000*
	JFJ6060*	JFJ6061*	JFJ6062*	JFJ6063*	JFJ6064*	JFJ6065*
	JFJ6066*	JFJ6067*	JFJ6068*			
JF06079	OURJOV*	JOVDMP*	JF06070	JF06071	JF06072	JF06073
	JF06074	JF06075	JF06076	JF06077	JF06078	JFOP000*
	JFJ6070*	JFJ6071*	JFJ6072*	JFJ6073*	JFJ6074*	JFJ6075*
	JFJ6076*	JFJ6077*	JFJ6078*			
JF06101	OURJOV*	JOVDMP*	JF06100	JFOP000*	JFJ6100*	
JF06109	OURJOV*	JOVDMP*	JF06108	JFOP000*	JFJ6108*	
JF06110	OURJOV*	JOVDMP*	JF06108	JFOP000*	JFJ6108*	
JF06111	OURJOV*	JOVDMP*	JF06108	JFOP000*	JFJ6108*	
JF06122	OURJOV*	JOVDMP*	JF06112	JF06113	JF06114	JF06115
	JF06116	JF06117	JF06118	JF06119	JFOP000*	JFJ6112*
	JFJ6113*	JFJ6114*	JFJ6115*	JFJ6116*	JFJ6117*	JFJ6118*
	JFJ6119*					
JF06140	OURJOV*	JOVDMP*	JF06132	JF06133	JF06134	JF06135
	JF06136	JF06137	JF06138	JF06139	JFOP000*	JFJ6132*
	JFJ6133*	JFJ6134*	JFJ6135*	JFJ6136*	JFJ6137*	JFJ6138*
	JFJ6139*					

* Source file supplied with USE file type only

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
JF06150	OURJOV*	JOVDMP*	JF06142	JF06143	JF06144	JF06145
	JF06146	JF06147	JF06148	JF06149	JF0P000*	JFJ6142*
	JFJ6143*	JFJ6144*	JFJ6145*	JFJ6146*	JFJ6147*	JFJ6148*
	JFJ6149*					
JF06160	OURJOV*	JOVDMP*	JF06152	JF06153	JF06154	JF06155
	JF06156	JF06157	JF06158	JF06159	JF0P000*	JFJ6152*
	JFJ6153*	JFJ6154*	JFJ6155*	JFJ6156*	JFJ6157*	JFJ6158*
	JFJ6159*					
JF06170	OURJOV*	JOVDMP*	JF06162	JF06163	JF06164	JF06165
	JF06166	JF06167	JF06168	JF06169	JF0P000*	JFJ6162*
	JFJ6163*	JFJ6164*	JFJ6165*	JFJ6166*	JFJ6167*	JFJ6168*
	JFJ6169*					
JF06180	OURJOV*	JOVDMP*	JF06172	JF06173	JF06174	JF06175
	JF06176	JF06177	JF06178	JF06179	JF0P000*	JFJ6172*
	JFJ6173*	JFJ6174*	JFJ6175*	JFJ6176*	JFJ6177*	JFJ6178*
	JFJ6179*					
JF06190	OURJOV*	JOVDMP*	JF06182	JF06183	JF06184	JF06185
	JF06186	JF06187	JF06188	JF06189	JF0P000*	JFJ6182*
	JFJ6183*	JFJ6184*	JFJ6185*	JFJ6186*	JFJ6187*	JFJ6188*
	JFJ6189*					
JF06199	OURJOV*	JOVDMP*	JOVSPC*	JF06191	JF06192	JF06193
	JF06194	JF06195	JF06196	JF06197	JF06198	JF0P000*
	JFJ6191*	JFJ6192*	JFJ6193*	JFJ6194*	JFJ6195*	JFJ6196*
	JFJ6197*	JFJ6198*				
JF0619I	OURJOV*	JOVDMP*	JOVSPC*	JF0619A**	JF0619B	JF0619C
	JF0619D	JF0619E	JF0619F	JF0619G	JF0619H	JF0P000*
	JFJ619A*	JFJ619B*	JFJ619C*	JFJ619D*	JFJ619E*	JFJ619F*
	JFJ619G*	JFJ619H*				
JF0619R	OURJOV*	JOVDMP*	JOVSPC*	JF0619J	JF0619K	JF0619L
	JF0619M	JF0619N	JF0619O	JF0619P	JF0619Q	JF0P000*
	JFJ619J*	JFJ619K*	JFJ619L*	JFJ619M*	JFJ619N*	JFJ619O*
	JFJ619P*	JFJ619Q*				
JFD6201	OURJOV*	JOVDMP*	JF0P000*			
JFD6209	OURJOV*	JOVDMP*	JF0P000*			
JFD6210	OURJOV*	JOVDMP*	JF0P000*			
JFD6211	OURJOV*	JOVDMP*	JF0P000*			
JFD6222	OURJOV*	JOVDMP*	JF0P000*	JFD6213	JFD6214	JFD6215
	JFD6216	JFD6217	JFD6218	JFD6219	JFJ6213*	JFJ6214*
	JFJ6215*	JFJ6216*	JFJ6217*	JFJ6218*	JFJ6219*	
JFD6233	OURJOV*	JOVDMP*	JF0P000*	JFD6223	JFD6224	JFD6225
	JFD6226	JFD6227	JFD6228	JFD6229	JFD6230	JFD6231
	JFD6232	JFJ6223*	JFJ6224*	JFJ6225*	JFJ6226*	JFJ6227*
	JFJ6228*	JFJ6229*	JFJ6230*	JFJ6231*	JFJ6232*	
JFD6253	OURJOV*	JOVDMP*	JF0P000*	JFD6243	JFD6244	JFD6245
	JFD6246	JFD6247	JFD6248	JFD6249	JFD6250	JFD6251
	JFD6252	JFJ6243*	JFJ6244*	JFJ6245*	JFJ6246*	JFJ6247*
	JFJ6248*	JFJ6249*	JFJ6250*	JFJ6251*	JFJ6252*	
JFD6269*	OURJOV*	JOVDMP*	JF0P000*	JFD6260*	JFD6261*	JFD6262*
	JFD6263*	JFD6264*	JFD6265*	JFD6266*	JFD6267*	JFD6268*
	JFJ6260**	JFJ6261**	JFJ6262**	JFJ6263**	JFJ6264**	JFJ6265**
	JFJ6266**	JFJ6267**	JFJ6268**			
JFD6279*	OURJOV*	JOVDMP*	JF0P000*	JFD6270*	JFD6271*	JFD6272*
	JFD6273*	JFD6274*	JFD6275*	JFD6276*	JFD6277*	JFD6278*
	JFJ6270**	JFJ6271**	JFJ6272**	JFJ6273**	JFJ6274**	JFJ6275**
	JFJ6276**	JFJ6277**	JFJ6278**			
JF06803	OURJOV*	JOVDMP*	JF06802	JF0P000*	JFJ6802*	
JF06807	OURJOV*	JOVDMP*	JF06806	JF0P000*	JFJ6806*	
JF06809	OURJOV*	JOVDMP*	JOVSPC*	JF06808	JF0P000*	JFJ6808*
JF0680B	OURJOV*	JOVDMP*	JOVSPC*	JF0680A	JF0P000*	JFJ680A*
JF0680D	OURJOV*	JOVDMP*	JOVSPC*	JF0680C	JF0P000*	JFJ680C*
JF0680F	OURJOV*	JOVDMP*	JOVSPC*	JF0680E	JF0P000*	JFJ680E*
JFD680H	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*	JFD680G	JFJ680G*

* Source file supplied with USE file type only

** Multiple versions of source file supplied(USE and JOV/CPL file types)

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
JFD680J	OURJOV*	JOVDMP*	JOVSPC*	JF0P000*	JFD680I	JFJ680I*
JF06811	OURJOV*	JOVDMP*	JF06810	JF0P000*	JFJ6810*	
JF06815	OURJOV*	JOVDMP*	JF06814	JF0P000*	JFJ6814*	
JF06817	OURJOV*	JOVDMP*	JF06816	JF0P000*	JFJ6816*	
JF06819	OURJOV*	JOVDMP*	JF06818	JF0P000*	JFJ6818*	
JF06821	OURJOV*	JOVDMP*	JF06820	JF0P000*	JFJ6820*	
JF06823	OURJOV*	JOVDMP*	JOVSPC*	JF06822*†	JF0P000*	JFJ6822*
JF06825	OURJOV*	JOVDMP*	JOVSPC*	JF06824	JF0P000*	JFJ6824*
JFDD600	OURJOV*	JOVSPC*	JF0P000*			
JFDD601	OURJOV*	JOVSPC*	JF0P000*			
JFDD602*†	OURJOV*	JOVSPC*	JF0P000*			
JFDD603*†	OURJOV*	JOVSPC*	JF0P000*			
JFDD604	OURJOV*	JOVSPC*	JF0P000*			
JFDD605	OURJOV*	JOVSPC*	JF0P000*			
JFDD606	OURJOV*	JOVSPC*	JF0P000*			
JFDD607	OURJOV*	JOVSPC*	JF0P000*			
JFDD608	OURJOV*	JOVSPC*	JF0P000*			
JFDD609	OURJOV*	JOVSPC*	JF0P000*			
JFDD610*†	OURJOV*	JOVSPC*	JF0P000*			
JFDD611*†	OURJOV*	JOVSPC*	JF0P000*			
JF0D720	OURJOV*					
JF0D721	OURJOV*					
JFDD722	OURJOV*	JOVTP*				
JFDD723	OURJOV*	JOVTP*				
JFDD724*	OURJOV*	JOVTP*				
JFDD725*	OURJOV*	JOVTP*				
JF0D727	OURJOV*					
JF0D728	OURJOV*	JOVSPC*				
JF0D729	OURJOV*	JOVSPC*				
JFDD72A	OURJOV*	JOVSPC*				
JFDD72B	OURJOV*	JOVSPC*				
JFDD72C	OURJOV*	JOVSPC*				
JF0D72D	OURJOV*	JOVSPC*				
JFDD72E	OURJOV*	JOVSPC*				
JFDD72F	OURJOV*	JOVSPC*				
JFDD72G	OURJOV*	JOVSPC*				
JFDD72H	OURJOV*	JOVSPC*				
JFDD72I	OURJOV*	JOVSPC*				
JFDDA01	OURJOV*	JOVSPC*	JF0P000*			
JFDDA02	OURJOV*	JOVSPC*	JF0P000*			
JFDF000	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JFDF001	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JFDF002	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JFDF003	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JFDF004	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JFDF005	OURJOV*	JOVDMP*	JOVMATH*	JF0P000*		
JGD0001	OURJOV*	JOVMATH*	JG0P000*	JGD0000*	JGF0001	
JGD0009	OURJOV*	JG0P000*	JGD0004*	JGD0005*	JGD0006*	JGD0007
	JGD0008*†					
JG0000C	OURJOV*	JOVDMP*	JG0P000*			
JGD000E	OURJOV*	JG0P000*	JGD0000*	JGD000D	JGF000E	JGJ000D*
JGN000K	OURJOV*	JG0P000*	JGN000F*	JGN000G*	JGN000H*	JGN000I
	JGN000J*†					
J000104	OURJOV*	JOVDMP*	J000002*	J000102	J000103	J00P000*
	JOJ0102*	JOJ0103*				
J000204	OURJOV*	JOVDMP*	J000202	J000203	J00P000*	JOJ0202*
	JOJ0203*					
J000300	OURJOV*	J00P000*				
J000305	OURJOV*	J00P000*				
J000310	OURJOV*	J00P000*				
J000313	OURJOV*	JOVDMP*	J000311	J000312	J00P000*	JOJ0311*
	JOJ0312*					

* Source file supplied with USE file type only

*† Multiple versions of source file supplied(USE and JOV/CPL file types)

Table G-6 JOVIAL Test Programs and Source Code Files (continued)

PROGRAM	SUPPORT FILES(* - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)					
J000316	OURJOV*	JOVDMP*	J000314	J000315	J00P000*	JOJ0314*
J000319*	JOJ0315*					
J000319*	OURJOV*	JOVDMP*	J000317*	J000318*	J00P000*	JOJ0317*
J000322*	JOJ0318*					
J000322*	OURJOV*	JOVDMP*	J000320*	J000321*	J00P000*	JOJ0320*
J000325	JOJ0321*					
J000325	OURJOV*	JOVDMP*	J000323	J000324	J00P000*	JOJ0323*
J000328	JOJ0324*					
J000328	OURJOV*	JOVDMP*	J000326	J000327	J00P000*	JOJ0326*
J000331	JOJ0327*					
J000331	OURJOV*	JOVDMP*	J000329	J000330	J00P000*	JOJ0329*
J000334	JOJ0330*					
J000334	OURJOV*	JOVDMP*	J000332	J000333	J00P000*	JOJ0332*
J000337	JOJ0333*					
J000337	OURJOV*	JOVDMP*	J000335	J000336	J00P000*	JOJ0335*
J000340	JOJ0336*					
J000340	OURJOV*	JOVDMP*	J000338	J000339	J00P000*	JOJ0338*
J000343	JOJ0339*					
J000343	OURJOV*	JOVDMP*	J000341	J000342	J00P000*	JOJ0341*
J000346	JOJ0342*					
J000346	OURJOV*	JOVDMP*	J00P000*	J000344	J000345	JOJ0344*
J000350	JOJ0345*					
J000350	OURJOV*	JOVDMP*	J000347	J000348	J000349	J00P000*
J000353	JOJ0347*	JOJ0348*	JOJ0349*			
J000353	OURJOV*	JOVDMP*	J000347	J000351	J000352	J00P000*
J000365	JOJ0347*	JOJ0351*	JOJ0352*			
J000365	OURJOV*	JOVDMP*	JOVSPC*	J000363	J000364	J00P000*
J00036K	JOJ0363*	JOJ0364*				
J00036K	OURJOV*	JOVDMP*	JOVSPC*	J00036I	J00036J	J00P000*
J00036N	JOJ036I*	JOJ036J*				
J00036N	OURJOV*	JOVDMP*	JOVSPC*	J00036L	J00036M	J00P000*
J00036Q	JOJ036L*	JOJ036M*				
J00036Q	OURJOV*	JOVDMP*	JOVSPC*	J00P000*	J00036O	J00036P
J00036T	JOJ036O*	JOJ036P*				
J00036T	OURJOV*	JOVDMP*	JOVSPC*	J00P000*	J00036R	J00036S
J00036H	JOJ036R*	JOJ036S*				
J00036H	OURJOV*	JOVDMP*	JOVSPC*	J00036U	J00036V	J00P000*
J00036Z	JOJ036U*	JOJ036V*				
J00036Z	OURJOV*	JOVDMP*	JOVTP*	J00P000*	J00036X	J00036Y
J000391	JOJ036X*	JOJ036Y*				
J000391	OURJOV*	JOVDMP*	J000390	J00P000*	JOJ0390*	
J000404	JOJ0403*		J000402	J000403	J00P000*	JOJ0402*
J000500	OURJOV*	J00P000*				
J000504	OURJOV*	JOVDMP*	J000502	J000503	J00P000*	JOJ0502*
J000509	JOJ0503*					
J000509	OURJOV*	JOVDMP*	J000507	J000508	J00P000*	JOJ0507*
J000514	JOJ0508*					
J000514	OURJOV*	JOVDMP*	J000512	J000513	J00P000*	JOJ0512*
J000519	JOJ0513*					
J000519	OURJOV*	JOVDMP*	J000517	J000518	J00P000*	JOJ0517*
J000525	JOJ0518*					
J000525	OURJOV*	JOVDMP*	J00P000*	J000523	J000524	JOJ0523*
J000604	JOJ0524*					
J000604	OURJOV*	JOVDMP*	J000602	J000603	J00P000*	JOJ0602*
J000605	JOJ0603*					
J000605	OURJOV*	JOVDMP*	J00P000*			
J000606	OURJOV*	JOVDMP*	J00P000*			
J000607	OURJOV*	JOVDMP*	J00P000*			
J000704	OURJOV*	JOVDMP*	J000702	J000703	J00P000*	JOJ0702*
J000705	JOJ0703*					
J000705	OURJOV*	JOVDMP*	J00P000*			
J000708	OURJOV*	JOVDMP*	J000706	J000707	J00P000*	JOJ0706*

* Source file supplied with USE file type only

Table G-6 JOVIAL Test Programs and Source Code Files (concluded)

<u>PROGRAM</u>	<u>SUPPORT FILES(*) - FILES NAMED IN COMPOOL AND COPY DIRECTIVES)</u>					
J000711	JOJ0707*		J000709	J000710	J00P000*	JOJ0709*
	OURJOV*	JOVDMP*				
	JOJ0710*					
J000713	OURJOV*	JOVDMP*	J00P000*			
J000714	OURJOV*	JOVDMP*	J00P000*			
J000715	OURJOV*	JOVDMP*	J00P000*			
J000716	OURJOV*	JOVDMP*	J00P000*			
J000717	OURJOV*	JOVDMP*	J00P000*			
J000718	OURJOV*	JOVDMP*	J00P000*			
J000721	OURJOV*	JOVDMP*	J000719	J000720	J00P000*	JOJ0719*
	JOJ0720*					

Table G-7. FORTRAN Test Programs and Source Code Files

<u>PROGRAM</u>	<u>SUPPORT FILES(* - FILES NAMED IN INCLUDE STATEMENTS)</u>					
FA00000	OURFOR*	FA0P000*				
FF03519	OURFOR*	FF03500	FF03502	FF03503	FF03504	FF03514
	FF0P000*					
FF03620	OURFOR*	FF03600	FF03602	FF03603	FF03605	FF03606
	FF03608	FF03609	FF0P000*			
FF04120	OURFOR*	FF0P000*				
FF04121	OURFOR*	FF0P000*				
FF04122	OURFOR*	FF0P000*				
FF04123	OURFOR*	FF0P000*				
FF04124	OURFOR*	FF0P000*				
FF04125	OURFOR*	FF0P000*				
FF04126	OURFOR*	FF0P000*				
FF04127	OURFOR*	FF0P000*				
FF04129	OURFOR*	FF0P000*				
FF04510	OURFOR*	FF0P000*				
FF04511	OURFOR*	FF0P000*				
FF04512	OURFOR*	FF0P000*				
FF04513	OURFOR*	FF0P000*				
FF04514	OURFOR*	FF0P000*				
FF04515	OURFOR*	FF0P000*				
FF04516	OURFOR*	FF0P000*				
FF04517	OURFOR*	FF0P000*				
FF0451J	OURFOR*	FF04518	FF04519	FF0451A	FF0451B	FF0451C
	FF0451D	FF0451E	FF0451F	FF0451G	FF0451H	FF0451I
	FF0P000*					
FFD451U*	OURFOR*	FORTYP*	FF0P000*	FFD451K	FFD451L	FFD451M
	FFD451N	FFD451O	FFD451P	FFD451Q	FFD451R	FFD451S
FF0451Z	OURFOR*	FF0451V	FF0451W	FF0451X	FF0451Y	FF0P000*
FF04520	OURFOR*	FF0P000*				
FF04521	OURFOR*	FF0P000*				
FF04522	OURFOR*	FF0P000*				
FFD4523	OURFOR*	FORTYP*	FF0P000*			
FFD4524	OURFOR*	FORTYP*	FF0P000*			
FFD4525	OURFOR*	FORTYP*	FF0P000*			
FFD4526	OURFOR*	FORTYP*	FF0P000*			
FF04529	OURFOR*	FF0P000*				
FF04530	OURFOR*	FF0P000*				
FF04531	OURFOR*	FF0P000*				
FF04532	OURFOR*	FF0P000*				
FF04533	OURFOR*	FF0P000*				
FF04534	OURFOR*	FF0P000*				
FF04535	OURFOR*	FF0P000*				
FF04536	OURFOR*	FF0P000*				
FF04537	OURFOR*	FF0P000*				
FF04538	OURFOR*	FF0P000*				
FF04539	OURFOR*	FF0P000*				
FF0453A	OURFOR*	FF0P000*				
FF04540	OURFOR*	FF0P000*				
FF04541	OURFOR*	FF0P000*				
FF04550	OURFOR*	FF0P000*				
FF04551	OURFOR*	FF0P000*				
FF04552	OURFOR*	FF0P000*				
FF04553	OURFOR*	FF0P000*				
FF04554	OURFOR*	FF0P000*				
FF04555	OURFOR*	FF0P000*				
FF04556	OURFOR*	FF0P000*				
FF04557	OURFOR*	FF0P000*				
FF04558	OURFOR*	FF0P000*				
FF04559	OURFOR*	FF0P000*				
FFD455D	OURFOR*	FORTYP*	FF0P000*			
FFD455E	OURFOR*	FORTYP*	FF0P000*			
FFD455F	OURFOR*	FORTYP*	FF0P000*			
FFD455G	OURFOR*	FORTYP*	FF0P000*			
FFD455I	OURFOR*	FORTYP*	FF0P000*			

* Source file supplied with USE file type only

Table G-7. FORTRAN Test Programs and Source Code Files (Continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - FILES NAMED IN INCLUDE STATEMENTS)</u>					
FFD455J	OURFOR*	FORTYP*	FF0P000*			
FFD455K	OURFOR*	FORTYP*	FF0P000*			
FFD455L	OURFOR*	FORTYP*	FF0P000*			
FF04560	OURFOR*	FF0P000*				
FF04562	OURFOR*	FF0P000*				
FF04563	OURFOR*	FF0P000*				
FFD4566*	OURFOR*	FORTYP*	FF0P000*			
FFD4567	OURFOR*	FORTYP*	FF0P000*			
FFD4568	OURFOR*	FORTYP*	FF0P000*			
FFD4569	OURFOR*	FORTYP*	FF0P000*			
FF04600	OURFOR*	FF0P000*				
FF04601	OURFOR*	FF0P000*				
FF04602	OURFOR*	FF0P000*				
FFD4603	OURFOR*	FORTYP*	FF0P000*			
FFD4604	OURFOR*	FORTYP*	FF0P000*			
FFD4605	OURFOR*	FORTYP*	FF0P000*			
FFD4606	OURFOR*	FORTYP*	FF0P000*			
FFD4607	OURFOR*	FORTYP*	FF0P000*			
FFD4608	OURFOR*	FORTYP*	FF0P000*			
FFD4609	OURFOR*	FORTYP*	FF0P000*			
FFD460A	OURFOR*	FORTYP*	FF0P000*			
FFD460B	OURFOR*	FORTYP*	FF0P000*			
FF0460I	OURFOR*	FF0P000*				
FF05200	OURFOR*	FF0P000*				
FF05201	OURFOR*	FF0P000*				
FF05202	OURFOR*	FF0P000*				
FF05203	OURFOR*	FF0P000*				
FF05204	OURFOR*	FF0P000*				
FF05205	OURFOR*	FF0P000*				
FF05206	OURFOR*	FF0P000*				
FF05207	OURFOR*	FF0P000*				
FF05208	OURFOR*	FF0P000*				
FF05209	OURFOR*	FF0P000*				
FF0520A	OURFOR*	FF0P000*				
FFD520C	OURFOR*	FORTYP*	FF0P000*			
FFD520D	OURFOR*	FORTYP*	FF0P000*			
FFD520E	OURFOR*	FORTYP*	FF0P000*			
FFD520F	OURFOR*	FORTYP*	FF0P000*			
FFD520G	OURFOR*	FORTYP*	FF0P000*			
FFD520H	OURFOR*	FORTYP*	FF0P000*			
FFD520I	OURFOR*	FORTYP*	FF0P000*			
FFD520J	OURFOR*	FORTYP*	FF0P000*			
FF05210	OURFOR*	FF0P000*				
FF05211	OURFOR*	FF0P000*				
FF05212	OURFOR*	FF0P000*				
FF05213	OURFOR*	FF0P000*				
FF05214	OURFOR*	FF0P000*				
FF05215	OURFOR*	FF0P000*				
FFD5218	OURFOR*	FORTYP*	FF0P000*			
FFD5219	OURFOR*	FORTYP*	FF0P000*			
FFD521A	OURFOR*	FORTYP*	FF0P000*			
FFD521B	OURFOR*	FORTYP*	FF0P000*			
FFD521C	OURFOR*	FORTYP*	FF0P000*			
FFD521D	OURFOR*	FORTYP*	FF0P000*			
FFD521E	OURFOR*	FORTYP*	FF0P000*			
FFD521F	OURFOR*	FORTYP*	FF0P000*			
FF0521H	OURFOR*	FF0P000*				
FF0521I	OURFOR*	FF0P000*				
FF0521J	OURFOR*	FF0P000*				
FF0521K	OURFOR*	FF0P000*				
FF05304	OURFOR*	FF05300	FF05301	FF05302	FF05303	FF0P000*
FF05308	OURFOR*	FF05305	FF05306	FF05307	FF0P000*	
FF05408	OURFOR*	FF05400	FF05401	FF05402	FF05403	FF05404

* Source file supplied with USE file type only

Table G-7. FORTRAN Test Programs and Source Code Files (Continued)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - FILES NAMED IN INCLUDE STATEMENTS)</u>					
FF05405	FF05406	FF05407	FF0P000*			
FF05505	OURFOR*	FF05501	FF05502	FF05503	FF05504	FF0P000*
FF0550C	OURFOR*	FF05506	FF05507	FF05508	FF05509	FF0550A
	FF0550B	FF0P000*				
FF0550D	OURFOR*	FF0P000*				
FF0550E	OURFOR*	FF0P000*				
FF0550F	OURFOR*	FF0P000*				
FF06001	OURFOR*	FF0P000*				
FF06009	OURFOR*	FF0P000*				
FF06010	OURFOR*	FF0P000*				
FF06011	OURFOR*	FF0P000*				
FF06022	OURFOR*	FF06013	FF06014	FF06015	FF06016	FF06017
	FF06018	FF06019	FF0P000*			
FF06033	OURFOR*	FF06023	FF06024	FF06025	FF06026	FF06027
	FF06028	FF06029	FF06030	FF06031	FF06032	FF0P000*
FF06053	OURFOR*	FF06043	FF06044	FF06045	FF06046	FF06047
	FF06048	FF06049	FF06050	FF06051	FF06052	FF0P000*
FF06069	OURFOR*	FF06060	FF06061	FF06062	FF06063	FF06064
	FF06065	FF06066	FF06067	FF06068	FF0P000*	
FF06079	OURFOR*	FF06070	FF06071	FF06072	FF06073	FF06074
	FF06075	FF06076	FF06077	FF06078	FF0P000*	
FF06101	OURFOR*	FF06100	FF0P000*			
FF06109	OURFOR*	FF06108	FF0P000*			
FF06110	OURFOR*	FF06108	FF0P000*			
FF06111	OURFOR*	FF06108	FF0P000*			
FF06122	OURFOR*	FF06112	FF06113	FF06114	FF06115	FF06116
	FF06117	FF06118	FF06119	FF0P000*		
FF06140	OURFOR*	FF06132	FF06133	FF06134	FF06135	FF06136
	FF06137	FF06138	FF06139	FF0P000*		
FF06150	OURFOR*	FF06142	FF06143	FF06144	FF06145	FF06146
	FF06147	FF06148	FF06149	FF0P000*		
FF06160	OURFOR*	FF06152	FF06153	FF06154	FF06155	FF06156
	FF06157	FF06158	FF06159	FF0P000*		
FF06170	OURFOR*	FF06162	FF06163	FF06164	FF06165	FF06166
	FF06167	FF06168	FF06169	FF0P000*		
FF06180	OURFOR*	FF06172	FF06173	FF06174	FF06175	FF06176
	FF06177	FF06178	FF06179	FF0P000*		
FF06190	OURFOR*	FF06182	FF06183	FF06184	FF06185	FF06186
	FF06187	FF06188	FF06189	FF0P000*		
FF06803	OURFOR*	FF06802	FF0P000*			
FF06807	OURFOR*	FF06806	FF0P000*			
FF06817	OURFOR*	FF06816	FF0P000*			
FF06819	OURFOR*	FF06818	FF0P000*			
FF06821	OURFOR*	FF06820	FF0P000*			
FF0D720	OURFOR*					
FF0D721	OURFOR*					
FFDD722	OURFOR*	FORTYP*				
FFDD723	OURFOR*	FORTYP*				
FFDD724	OURFOR*	FORTYP*				
FFDD725	OURFOR*	FORTYP*				
FF0D727	OURFOR*					
FF0D728	OURFOR*					
FFDE220	OURFOR*	FF0P000*	FFDE000*	FFDE201	FFDE202	FFDE203
	FFDE204	FFDE205	FFDE206	FFDE207	FFDE208	FFDE209
	FFDE20A	FFDE211	FFDE212	FFDE213	FFDE214	FFDE215
	FFDE216					
FFDE250	OURFOR*	FF0P000*	FFDE000*	FFDE231	FFDE232	FFDE233
	FFDE234	FFDE235	FFDE236	FFDE237	FFDE238	FFDE239
	FFDE23A	FFDE241	FFDE242	FFDE243	FFDE244	FFDE245
	FFDE246					
FFDE420	OURFOR*	FF0P000*	FFDE000*	FFDE401	FFDE402	FFDE403
	FFDE404	FFDE405	FFDE406	FFDE407	FFDE408	FFDE409

Table G-7. FORTRAN Test Programs and Source Code Files (Concluded)

<u>PROGRAM</u>	<u>SUPPORT FILES(* - FILES NAMED IN INCLUDE STATEMENTS)</u>					
	FFDE40A	FFDE411	FFDE412	FFDE413	FFDE414	FFDE415
	FFDE416					
FFDE450	OURFOR*	FF0P000*	FFDE000*	FFDE431	FFDE432	FFDE433
	FFDE434	FFDE435	FFDE436	FFDE437	FFDE438	FFDE439
	FFDE43A	FFDE441	FFDE442	FFDE443	FFDE444	FFDE445
	FFDE446					
FFDF000	OURFOR*	FF0P000*				
FFDF001	OURFOR*	FF0P000*				
FFDF002	OURFOR*	FF0P000*				
FFDF003	OURFOR*	FF0P000*				
FFDF004	OURFOR*	FF0P000*				
FFDF005	OURFOR*	FF0P000*				
FGD0001	OURFOR*	FG0P000*	FGD0000*			
FGD000E	OURFOR*	FG0P000*	FGD0000*	FGD000D		
F000104	OURFOR*	F000002*	F000102	F000103	F00P000*	
F000204	OURFOR*	F000202	F000203	F00P000*		
F000300	OURFOR*	F00P000*				
F000305	OURFOR*	F00P000*				
F000310	OURFOR*	F00P000*				
F000313	OURFOR*	F000311	F000312	F00P000*		
F000316	OURFOR*	F000314	F000315	F00P000*		
F000319	OURFOR*	F000317	F000318	F00P000*		
F000322	OURFOR*	F000320	F000321	F00P000*		
F000325	OURFOR*	F000323	F000324	F00P000*		
F000328	OURFOR*	F000326	F000327	F00P000*		
F000331	OURFOR*	F000329	F000330	F00P000*		
F000334	OURFOR*	F000332	F000333	F00P000*		
F000337	OURFOR*	F000335	F000336	F00P000*		
F000340	OURFOR*	F000338	F000339	F00P000*		
F000343	OURFOR*	F000341	F000342	F00P000*		
F0D0346	OURFOR*	F00P000*	F0D0344	F0D0345		
F000350	OURFOR*	F000347	F000348	F000349	F00P000*	
F000353	OURFOR*	F000347	F000351	F000352	F00P000*	
F000356	OURFOR*	F000354	F000355	F00P000*		
F000359	OURFOR*	F000357	F000358	F00P000*		
F000362	OURFOR*	F000360	F000361	F00P000*		
F0D0362	OURFOR*					
F000391	OURFOR*	F000390	F00P000*			
F000404	OURFOR*	F000402	F000403	F00P000*		
F000500	OURFOR*	F00P000*				
F000504	OURFOR*	F000502	F000503	F00P000*		
F000509	OURFOR*	F000507	F000508	F00P000*		
F000514	OURFOR*	F000512	F000513	F00P000*		
F000519	OURFOR*	F000517	F000518	F00P000*		
F000604	OURFOR*	F000602	F000603	F00P000*		
F000605	OURFOR*	F00P000*				
F000606	OURFOR*	F00P000*				
F000607	OURFOR*	F00P000*				
F000704	OURFOR*	F000702	F000703	F00P000*		
F000705	OURFOR*	F00P000*				
F000708	OURFOR*	F000706	F000707	F00P000*		
F000711	OURFOR*	F000709	F000710	F00P000*		
F000713	OURFOR*	F00P000*				
F000714	OURFOR*	F00P000*				
F000715	OURFOR*	F00P000*				
F000716	OURFOR*	F00P000*				
F000717	OURFOR*	F00P000*				
F000718	OURFOR*	F00P000*				
F000721	OURFOR*	F000719	F000720	F00P000*		

Table G-8 Ada Type A Test Program Input/Output Files

<u>PROGRAM</u>	<u>INPUT/OUTPUT FILES</u>				
AFDE220	OFDE201 OFDE206 OFDE211 OFDE216	OFDE202 OFDE207 OFDE212	OFDE203 OFDE208 OFDE213	OFDE204 OFDE209 OFDE214	OFDE205 OFDE20A OFDE215
AFDE250	IFDE231 IFDE236 IFDE241 IFDE246	IFDE232 IFDE237 IFDE242	IFDE233 IFDE238 IFDE243	IFDE234 IFDE239 IFDE244	IFDE235 IFDE23A IFDE245
AFDE420	OFDE401 OFDE406 OFDE411 OFDE416	OFDE402 OFDE407 OFDE412	OFDE403 OFDE408 OFDE413	OFDE404 OFDE409 OFDE414	OFDE405 OFDE40A OFDE415
AFDE450	IFDE431 IFDE436 IFDE441 IFDE446	IFDE432 IFDE437 IFDE442	IFDE433 IFDE438 IFDE443	IFDE434 IFDE439 IFDE444	IFDE435 IFDE43A IFDE445
ALDE2C1	OLE2001				
ALDE2C2	OLE2001 OLE2006	OLE2002 OLE2007	OLE2003 OLE2008	OLE2004 OLE2009	OLE2005 OLE2010
ALDE2C3	OLE2001				
ALDE2C4	OLE2001 OLE2006	OLE2002 OLE2007	OLE2003 OLE2008	OLE2004 OLE2009	OLE2005 OLE2010
ALDE2C5	ILE2001				
ALDE2C6	ILE2001 ILE2006	ILE2002 ILE2007	ILE2003 ILE2008	ILE2004 ILE2009	ILE2005 ILE2010
ALDE2C7	ILE2001				
ALDE2C8	ILE2001 ILE2006	ILE2002 ILE2007	ILE2003 ILE2008	ILE2004 ILE2009	ILE2005 ILE2010
ALDE4C1	OLE4001				
ALDE4C2	OLE4001 OLE4006	OLE4002 OLE4007	OLE4003 OLE4008	OLE4004 OLE4009	OLE4005 OLE4010
ALDE4C3	OLE4001				
ALDE4C4	OLE4001 OLE4006	OLE4002 OLE4007	OLE4003 OLE4008	OLE4004 OLE4009	OLE4005 OLE4010
ALDE4C5	ILE4001				
ALDE4C6	ILE4001 ILE4006	ILE4002 ILE4007	ILE4003 ILE4008	ILE4004 ILE4009	ILE4005 ILE4010
ALDE4C7	ILE4001				
ALDE4C8	ILE4001 ILE4006	ILE4002 ILE4007	ILE4003 ILE4008	ILE4004 ILE4009	ILE4005 ILE4010